

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

# การศึกษาเทคโนโลยีคลังข้อมูลเพื่อใช้วิเคราะห์การขาย

## A Study of Applying Data Warehouse for Sales Analysis



\*H002890\*



วัน เดือน ปี.....	1 9 ๒๕๕๖
เลขทะเบียน.....	02890
เลขเรียกหนังสือ.....	๑๗๖ ๗ ๕๙2๗ 2545
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

1/9/56  
112

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการศึกษาระดับพิเศษ  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคเรียนที่ 1 ปีการศึกษา 2545  
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การศึกษาเทคโนโลยีคลังข้อมูลเพื่อใช้วิเคราะห์การขาย
นักศึกษา	นางสาวบุษกร กนกบรรณกร
อาจารย์ที่ปรึกษา	ดร. ภัทรชัย ลลิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	การจัดการเทคโนโลยีสารสนเทศ
ปีการศึกษา	2545

## บทคัดย่อ

ปัจจุบันระบบสารสนเทศได้เข้ามามีบทบาทสำคัญในโลกธุรกิจต่างๆ มากมาย ไม่ว่าจะเป็นทั้งในหน่วยงานภาครัฐหรือเอกชน องค์กรต่างๆ ได้พากันหันมาให้ความสนใจในการนำทรัพยากรข้อมูลที่มีอยู่อย่างมากมายทั้งจากภายนอกและภายในองค์กรมาใช้ให้เป็นประโยชน์ต่อการดำเนินธุรกิจ ด้วยความสามารถของเทคโนโลยีคลังข้อมูล ในการจัดการกับข้อมูลปริมาณมากๆ แล้ววิเคราะห์สรุปเป็นข้อมูลสำหรับระดับบริหารเพื่อนำมาใช้ประกอบการตัดสินใจ จึงนับเป็นอีกเทคโนโลยีหนึ่งที่องค์กรต่างๆ ให้ความสนใจนำมาใช้ร่วมกันกับระบบฐานข้อมูลปฏิบัติงานประจำวันเพื่อส่งเสริมให้การดำเนินงานขององค์กรมีประสิทธิภาพมากยิ่งขึ้น ซึ่งจะส่งผลต่อการเติบโตและก้าวหน้าทางธุรกิจขององค์กรต่อไป การศึกษานี้ได้นำเสนอหลักการและทฤษฎีของเทคโนโลยีคลังข้อมูล แนวทางในการสร้างคลังข้อมูล การใช้เครื่องมือสำหรับช่วยในการวิเคราะห์ข้อมูลจากคลังข้อมูล ตลอดจนได้นำเสนอกรณีศึกษาการสร้างคลังข้อมูลสำหรับวิเคราะห์การขายสินค้าเครื่องประดับทองคำ 99.99% ของบริษัทโกลด์มาสเตอร์จำกัด (มหาชน)

<b>Title</b>	A Study of Applying Data Warehouse for Sales Analysis
<b>Student</b>	Miss Boosakorn Kanokbannakorn
<b>Advisor</b>	Dr. Pattarachai Lalitrojwong
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Technology Management
<b>Academic Year</b>	2002

## ABSTRACT

An Informaion systems has emerged as an essential business entity. Over the years, organizations have been accumulating large amount of data which contain potentially valuable knowledge about their business but these data are usually not easily accessible for management and decision support. Data warehousing technology is a collection of decision support technologies, aimed at enabling knowledge worker to make better and faster decision. It can manipulate a large amount of data with the speed of information delivery and analysis. So it is gaining wide acceptance. Many organizations apply data warehouse to use with their legacy or operational system to improve their performance and business growth. This project describes an overview of data warehousing, presents front end client tools for querying and data analysis and back end tools for extracting, cleaning and loading data into a data warehouse. Finally, it illustrates a case study of applying data warehouse for sales analysis.

## กิตติกรรมประกาศ

ในการจัดทำโครงการศึกษาระณีพิเศษเรื่อง การศึกษาเทคโนโลยีคลังข้อมูลเพื่อใช้วิเคราะห์ การขาย ผู้เขียนต้องขอขอบพระคุณ ดร. ภัทรชัย กลิตโรจน์วงศ์ อาจารย์ที่ปรึกษาที่กรุณาให้คำ ปรึกษาและให้คำแนะนำในการจัดทำ ขอขอบคุณ คุณสมิทธิกรณ บัญยะไพโรเรื่อง ที่ช่วยจัดหา ซอฟต์แวร์ คู่มือการใช้งานและจัดหาหนังสือเพื่อประกอบการค้นคว้า และขอขอบคุณไปยังทาง บริษัท โกลด์มาสเตอร์ จำกัด (มหาชน) ที่ได้ให้การสนับสนุนด้านข้อมูลของบริษัทเป็นอย่างดี ตลอดจน ทุกท่านที่มีส่วนร่วมทำให้โครงการฉบับนี้สำเร็จไปได้ด้วยดี

บุษกร กนกบรรณกร



# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	2
1.3 ขั้นตอนการศึกษา.....	2
1.4 ขอบเขตการศึกษา.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. คลังข้อมูล (Data Warehouse).....	4
2.1 แนวคิดเกี่ยวกับคลังข้อมูล.....	4
2.2 ระบบฐานข้อมูลองค์กรกับคลังข้อมูล.....	4
2.3 ความหมายของคลังข้อมูล.....	7
2.4 จุดประสงค์ของคลังข้อมูล.....	8
2.5 เทคโนโลยีที่สนับสนุนคลังข้อมูล.....	8
2.6 Data Mart กับ Data Warehouse.....	9
3. สถาปัตยกรรมของคลังข้อมูล.....	12
3.1 สถาปัตยกรรมของคลังข้อมูล.....	12
3.2 สถาปัตยกรรมอื่นที่เกี่ยวข้องในการสร้างคลังข้อมูล.....	16
3.3 สถาปัตยกรรมของโครงสร้างข้อมูล.....	16
3.4 สถาปัตยกรรมของเซิร์ฟเวอร์.....	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5	สถาปัตยกรรมของระบบจัดการฐานข้อมูล.....	19
3.6	การเลือกสถาปัตยกรรมเพื่อสนับสนุนการทำงานของคลังข้อมูล .....	22
4.	เครื่องมือวิเคราะห์ข้อมูล.....	25
4.1	เครื่องมือในการสอบถามและทำรายงาน .....	25
4.2	On-line Analytical Processing (OLAP) .....	26
4.3	Executive Information Systems (EIS).....	28
4.4	Data Mining .....	28
4.5	แนวทางการเลือกเครื่องมือวิเคราะห์ข้อมูล.....	30
5.	การออกแบบฐานข้อมูลสำหรับคลังข้อมูล.....	32
5.1	การออกแบบฐานข้อมูลด้วย Star Schema .....	32
5.2	การออกแบบฐานข้อมูลด้วย Snowflake Schema.....	34
6.	แนวทางการพัฒนาคลังข้อมูล .....	35
6.1	ขั้นตอนการออกแบบและพัฒนาคลังข้อมูล.....	35
6.2	บทบาทของเครื่องมือช่วยในกระบวนการ ETL .....	38
6.3	เครื่องมือสำหรับทำ Data Profiling.....	40
6.4	ประโยชน์ของการทำ Data Profiling .....	40
7.	การศึกษาเครื่องมือที่ใช้วิเคราะห์ข้อมูลในคลังข้อมูล.....	42
7.1	บทบาทของ Microsoft SQL Server Analysis Services.....	42
7.2	การทำงานของ Analysis Services .....	43
8.	กรณีศึกษาการพัฒนาระบบคลังข้อมูลสำหรับวิเคราะห์การขาย.....	45
8.1	ขั้นตอนการดำเนินงาน .....	45
8.2	การศึกษารายละเอียดและโครงสร้างฐานข้อมูลประจำวันเดิม .....	46
8.3	การวิเคราะห์และออกแบบฐานข้อมูลสำหรับคลังข้อมูล .....	46
8.4	การอัปเดตข้อมูลในคลังข้อมูล.....	48
8.5	การสร้างและออกแบบ Model.....	50
8.6	การวิเคราะห์ข้อมูลโดย Microsoft Excel 2000 PivotTable .....	56
9.	บทสรุป .....	60
9.1	สรุปผลการศึกษา.....	60
9.2	ข้อเสนอแนะ.....	60

บรรณานุกรม .....	61
ภาคผนวก.....	62
ประวัติผู้เขียน.....	71



## สารบัญตาราง

	หน้า
ตารางที่	
2.1 เปรียบเทียบระหว่าง Operational System กับ Data Warehouse .....	6
2.2 เปรียบเทียบระหว่าง Data Mart กับ Data Warehouse.....	10
3.1 เปรียบเทียบระหว่าง DBMS ชนิดต่างๆ .....	22
3.2 Data Warehouse Decision Matrix.....	24
4.1 เปรียบเทียบการเลือกเครื่องมือวิเคราะห์ข้อมูลประเภทต่างๆ .....	31



# สารบัญภาพ

ภาพที่	หน้า
3.1 Data Warehousing Architecture.....	12
3.2 สถาปัตยกรรม Data Warehouse แบบรวมศูนย์.....	17
3.3 Federated Architecture.....	17
3.4 Tiered Architecture.....	18
3.5 สถาปัตยกรรมของเซิร์ฟเวอร์.....	19
3.6 Multidimensional Data.....	20
4.1 ระบบ Data Mining กับการนำไปใช้งาน.....	30
5.1 Star Schema.....	33
5.2 Snowflake Schema.....	34
6.1 ขั้นตอนการพัฒนาคลังข้อมูล.....	35
8.1 ER-Diagram ของฐานข้อมูลประจำวันการขายสินค้า.....	46
8.2 Snowflake Schema for Sales.....	48
8.3 หน้าจอแสดงฟอร์มการอัปเดตข้อมูลในคลังข้อมูล.....	48
8.4 หน้าจอแสดงการเรียกใช้แบบสอบถามแบบผนวกข้อมูล.....	49
8.5 หน้าจอแสดงการยืนยันจำนวนแถวที่จะทำการผนวกข้อมูล.....	49
8.6 หน้าจอแสดงผลการผนวกข้อมูลที่สำเร็จแล้ว.....	49
8.7 การเลือกลำดับขั้นของวันที่โดย Dimension Wizard.....	52
8.8 หน้าจอแสดงวิธีเลือกการประมวลผลคิวบ์.....	53
8.9 หน้าจอแสดงรายละเอียดการประมวลผลคิวบ์.....	54
8.10 หน้าจอแสดง Cube Browser.....	55
8.11 Cube Browser แสดงยอดขายสินค้าประเภทเข็มไทด์ราคา 3,000-4,999 บาท.....	55
8.12 Cube Browser แสดงยอดขายสินค้าเทียบกับเป้า ใน ไตรมาสที่ 4 ปี 2001.....	56
8.13 รายงานยอดขายแยกตามประเภทและคอลเลกชันสินค้าของทุกปี.....	57
8.14 กราฟแสดงแนวโน้มยอดขายเฉพาะที่ห้างเซ็นทรัล สาขาชิดลม.....	58

8.15 รายงานยอดขายสินค้าทุกสาขา แยกเป็นรายเดือน .....	58
8.16 กราฟเปรียบเทียบยอดขายปี 2000 ของพนักงานขายทุกคน .....	59
8.17 กราฟเปรียบเทียบยอดขาย ปี 2001 ของทุกสาขา .....	59



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเป็นยุคแห่งข้อมูลข่าวสาร ข้อมูลหรือสารสนเทศถือว่าเป็นสิ่งสำคัญที่สุดขององค์กรต่างๆ เนื่องจากข้อมูลหรือสารสนเทศที่เกี่ยวข้องกับการดำเนินงานขององค์กรจะเป็นตัวตัดสินใจชะตาความอยู่รอดหรือความล้มเหลวขององค์กรได้ เพราะว่าการดำเนินงานขององค์กรจะบรรลุเป้าหมายได้หรือไม่ขึ้นอยู่กับการปฏิบัติงานของบุคลากรของแต่ละแผนก แต่ละฝ่าย และผู้บริหารทุกระดับขององค์กร ซึ่งบุคลากรเหล่านั้นจะปฏิบัติงานโดยขาดข้อมูลหรือสารสนเทศที่เกี่ยวข้องกับการปฏิบัติงานไม่ได้ เพราะก่อนที่แต่ละคนจะลงมือปฏิบัติทำอะไรก็แล้วแต่ต้องมีการตัดสินใจก่อนทำเสมอ ซึ่งถ้าตัดสินใจถูกการปฏิบัติก็จะส่งผลทางบวก แต่ถ้าตัดสินใจผิดการปฏิบัติก็จะส่งผลทางลบ ดังนั้น การนำข้อมูลหรือสารสนเทศมาที่เก็บไว้ มาใช้เพื่อสนับสนุนการตัดสินใจหรือวิเคราะห์ให้เกิดผลทางธุรกิจจึงถือว่าเป็นสิ่งสำคัญอย่างยิ่งสำหรับองค์กร อีกทั้งปัจจุบันเทคโนโลยีสารสนเทศมีการเติบโตและพัฒนาไปอย่างรวดเร็วตลอดจนได้เข้ามามีบทบาทสำคัญในการประกอบธุรกิจต่างๆ มากมาย ทำให้การพัฒนาระบบงานภายในองค์กรทำได้ง่ายและมีประสิทธิภาพสูงกว่าสมัยก่อนเป็นอย่างมาก ประกอบกับสภาพการแข่งขันในการดำเนินธุรกิจ บริษัททุกแห่งต่างต้องการตอบสนองความต้องการของลูกค้าให้ดีขึ้น ต้องการเพิ่มผลกำไรและเพิ่มส่วนแบ่งตลาดของบริษัทให้มากขึ้นด้วย บริษัทที่มีข้อมูลเหนือกว่า ถูกต้องและแม่นยำกว่า ย่อมได้เปรียบในการนำข้อมูลมาจัดการวางแผนกลยุทธ์ทางธุรกิจหรือกำหนดทิศทางของธุรกิจ เพื่อเป็นการสร้างโอกาสทางการค้า ดังนั้นองค์กรต่างๆ จึงได้พยายามที่จะนำเทคโนโลยีใหม่ๆ เข้ามาช่วยในการพัฒนาระบบงานเพื่อให้ได้มาซึ่งข้อมูลที่เหนือกว่าในทุกๆ ด้าน และทำให้การทำงานขององค์กรเป็นไปโดยสะดวกรวดเร็วและมีประสิทธิภาพมากที่สุด

เทคโนโลยีคลังข้อมูลหรือ Data Warehouse เป็นอีกเทคโนโลยีหนึ่งที่ได้เข้ามามีบทบาทในการใช้ควบคุมข้อมูลและบริหารข้อมูลให้เกิดประโยชน์สูงสุด กล่าวคือข้อมูลขององค์กรที่มีอยู่นับวันจะมีมากขึ้นจนเป็นปัญหาสำหรับองค์กรที่จะต้องจัดการเก็บข้อมูลจำนวนมากเหล่านั้นมาใช้ให้เกิดประโยชน์ต่อองค์กร ในอดีตที่ผ่านมามีองค์กรหลายแห่งที่มีข้อมูลเป็นจำนวนมาก แต่ไม่สามารถใช้ข้อมูลในฐานะข้อมูลที่มีอยู่ให้เกิดประโยชน์สูงสุดได้ Data Warehouse จึงเป็นเครื่องมือหรือระบบที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะช่วยให้สามารถสร้างฐานข้อมูลความรู้จากฐานข้อมูลเดิมที่มีอยู่เพื่อนำมาใช้ช่วยสนับสนุนการตัดสินใจของผู้บริหารให้เป็นไปอย่างมีทิศทางที่แม่นยำมากขึ้น ทำให้สร้างแผนการดำเนินงานธุรกิจได้ดียิ่งขึ้น และช่วยให้ผู้บริหารสามารถเข้าถึงข้อมูลในองค์กรได้ทุกระดับ

### 1.2 วัตถุประสงค์ของการศึกษา

ในการศึกษาเกี่ยวกับเทคโนโลยีคลังข้อมูลมีวัตถุประสงค์ในการศึกษา ดังนี้

1. เพื่อเป็นแนวทางในการสร้างคลังข้อมูลสำหรับองค์กร
2. เพื่อก่อให้เกิดการนำข้อมูลในองค์กรมาใช้ให้เกิดประโยชน์สูงสุด
3. เพื่อเป็นการจัดทำข้อมูลใช้สำหรับสนับสนุนการตัดสินใจของผู้บริหาร
4. เพื่อช่วยให้ผู้บริหารสามารถนำข้อมูลที่มีอยู่มาใช้ในการวางแผนบริหารการขยายได้อย่างมีประสิทธิภาพ

### 1.3 ขั้นตอนการศึกษา

ในการศึกษาเกี่ยวกับการออกแบบและพัฒนาคลังข้อมูลมีขั้นตอนการดำเนินงานตามระยะเวลา ดังนี้

<b>ขั้นตอนการศึกษา</b>	<b>ม.ย. 45</b>	<b>ก.ค. 45</b>	<b>ส.ค. 45</b>	<b>ก.ย. 45</b>
------------------------	----------------	----------------	----------------	----------------

1. ศึกษาแนวคิดเกี่ยวกับคลังข้อมูล

[Redacted]

2. ศึกษาทฤษฎีการสร้างคลังข้อมูล

[Redacted]

3. ศึกษาเครื่องมือที่ใช้สร้างคลังข้อมูล

[Redacted]

4. ศึกษาโครงสร้างฐานข้อมูลของระบบงานขาย

[Redacted]

5. จัดทำระบบคลังข้อมูลเพื่อใช้วิเคราะห์การขาย

[Redacted]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 1.4 ขอบเขตการศึกษา

การศึกษาเกี่ยวกับออกแบบและสร้างคลังข้อมูลเพื่อใช้วิเคราะห์การขาย มีขอบเขตในการศึกษา ดังนี้

1. ศึกษาแนวคิดเกี่ยวกับคลังข้อมูล
2. ศึกษาทฤษฎีการออกแบบที่เกี่ยวข้องกับการสร้างคลังข้อมูล
3. ศึกษาเครื่องมือที่ใช้ในการสร้างคลังข้อมูลและที่ใช้ในการวิเคราะห์ข้อมูล ในที่นี้เลือกศึกษาการใช้งานของ Microsoft SQL Server 2000 Analysis Services
4. กรณีศึกษาการสร้างคลังข้อมูลเพื่อใช้วิเคราะห์การขายเครื่องประดับทองคำ 99.99% ของบริษัทโกลด์มาสเตอร์จำกัด (มหาชน)
  - ศึกษาโครงสร้างฐานข้อมูลของระบบงานขายสินค้าเครื่องประดับทองคำ
  - ออกแบบและสร้างระบบคลังข้อมูลเพื่อใช้วิเคราะห์การขาย
  - วิเคราะห์ข้อมูลสรุปในการขาย

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เป็นการนำแนวคิดของการจัดทำคลังข้อมูลมาประยุกต์ใช้กับงานปัจจุบันในองค์กรที่ทำงานอยู่
2. เป็นการนำข้อมูลที่จัดเก็บอยู่ฐานข้อมูลที่มีอยู่เดิมมาปรับใช้เพื่อตอบสนองความต้องการของผู้ใช้ได้เหมาะสม
3. เป็นการนำข้อมูลมาวิเคราะห์เพื่อหาแนวโน้ม และทำรายงานต่างๆตามที่ผู้บริหารต้องการได้หลายๆ รูปแบบ โดยไม่ต้องรอโปรแกรมจากโปรแกรมเมอร์ ทำให้การเตรียมข้อมูลเพื่อใช้สนับสนุนการตัดสินใจของผู้บริหารเป็นไปได้โดยสะดวกรวดเร็วและทันต่อการใช้งาน
4. ทำให้การทำงานและการบริหารข้อมูลภายในองค์กรเป็นไปได้เป็นอย่างดีมีประสิทธิภาพมากขึ้น

## บทที่ 2

### คลังข้อมูล (Data Warehouse)

#### 2.1 แนวคิดเกี่ยวกับคลังข้อมูล

ในทุกองค์กรต่างต้องการผลผลิตด้านธุรกิจของตนให้พุ่งไปข้างหน้า การรวบรวมข้อมูลเพื่อใช้ในการบริหารนับว่าเป็นปัจจัยสำคัญประการหนึ่ง หากแต่ว่าการเข้าใจข้อมูลเหล่านั้นและการนำข้อมูลเหล่านั้นมาใช้ได้อย่างมีประสิทธิภาพก็นับเป็นปัจจัยที่สำคัญยิ่งกว่า ปัญหาขององค์กรส่วนใหญ่ที่เมื่อผู้บริหารต้องการข้อมูลเพื่อใช้สนับสนุนการตัดสินใจ ยกตัวอย่างเช่น ต้องการทราบข้อมูลจำนวนลูกค้าของบริษัทที่อยู่ในช่วงอายุระหว่าง 20-30 ปี ตลอดเวลา 5 ปีที่ผ่านมา หรือต้องการทราบข้อมูลสินค้าที่ทำรายได้ดีที่สุดที่คนนับเฉพาะลูกค้าผู้หญิง ในช่วง 10 ปีที่ผ่านมา ซึ่งข้อมูลดังกล่าวไม่สามารถหาได้จากระบบฐานข้อมูลที่มีอยู่ ซึ่งหากต้องการข้อมูลสรุปเหล่านั้น ปัญหาที่จะพบก็คือบริษัทต้องทำการรวบรวมข้อมูลทั้งหมดที่กระจัดกระจายอยู่คนละเครื่องคนละแพลตฟอร์ม ข้อมูลคนละฟอร์แมต และสื่อที่ใช้ในการถ่ายข้อมูลระหว่างเครื่องก็ยังเป็นมาตรฐานคนละชนิดกันอีก กับอีกประเด็นที่สำคัญก็คือฐานข้อมูลที่มีอยู่นั้น ไม่ได้ถูกออกแบบมาเพื่อสนับสนุนข้อมูลดังกล่าว หากพยายามจะใช้งานไปทางนั้น จะพบว่าจะเกิดผลกระทบทำให้ระบบโดยรวมช้าลงไปหมด ดังนั้นถ้ามีระบบตรงกลางเกิดขึ้นมารวบรวมข้อมูลเหล่านั้นเข้ามาเก็บไว้สำหรับหาข้อมูลเพื่อสนับสนุนการตัดสินใจ สิ่งที่บริษัทจะได้รับก็มากมายเกินคุ้ม บริษัทจะเข้าใจพฤติกรรมลูกค้าได้ดีขึ้น เจาะกลุ่มลูกค้าที่ทำผลกำไรให้บริษัทได้มากที่สุด และมองข้อมูลได้ในหลายแง่มุมที่ไม่เคยมองเห็นมาก่อน ระบบตรงกลางนี้เราเรียกว่าคลังข้อมูล หรือ Data Warehouse

#### 2.2 ระบบฐานข้อมูลองค์กรกับคลังข้อมูล

ระบบฐานข้อมูลขององค์กรเป็นส่วนที่มีความสำคัญยิ่งในการวางรากฐานให้กับองค์กร โดยฐานข้อมูลจะรวบรวมข้อมูลที่เกี่ยวข้องในการดำเนินขององค์กรไว้ในลักษณะที่สอดคล้องกับความต้องการและโครงสร้างขององค์กร ซึ่งข้อมูลที่บันทึกลงในฐานข้อมูลนั้นประกอบด้วยข้อมูลจากทั้งภายนอกและภายในองค์กร ข้อมูลทั้งในอดีตและปัจจุบัน จากข้อมูลดังกล่าวถึงแม้ว่าจะมีอยู่ในระบบฐานข้อมูลก็ตาม แต่การเรียกใช้ข้อมูลเพื่อสนับสนุนการตัดสินใจจากฐานข้อมูลนั้นก็ใช้เวลาสูง และสิ้นเปลืองค่าใช้จ่ายทั้งด้านบุคลากรและการลงทุน ผู้บริหารไม่สามารถเรียกดูข้อมูลใน

รูปแบบที่ต้องการได้ด้วยตนเอง ต้องให้กลุ่มผู้เชี่ยวชาญหรือเจ้าหน้าที่คอมพิวเตอร์เป็นผู้ดำเนินการให้ทุกครั้งไป อีกทั้งการดึงข้อมูลก็จำเป็นต้องอาศัยข้อมูลที่เก็บอยู่ในหลายๆ เครื่อง หลายๆ ฐานข้อมูล หลายๆ สถานที่ ในเครือข่ายคอมพิวเตอร์ เนื่องจากเครื่องบางเครื่องสามารถเรียกดูข้อมูลได้เฉพาะข้อมูลบางเรื่องบางส่วนเท่านั้น ไม่สามารถเรียกดูข้อมูลตามต้องการได้ทั้งหมดจากเครื่องเดียวกัน อีกทั้งข้อมูลยังมีข้อจำกัดอยู่ คือต้องเป็นข้อมูลชนิด Text Base เท่านั้น ไม่อาจเรียกดูข้อมูลชนิดมัลติมีเดียที่มีภาพและเสียงได้ จากความสำคัญดังกล่าวของระบบฐานข้อมูล จึงได้มีการศึกษาถึงระบบการจัดการฐานข้อมูล และวิธีการที่สามารถเข้าถึงข้อมูลได้จากที่ใดก็ได้ ไม่ว่าจะข้อมูลจะเป็นแบบใดก็ตาม และเรียกดูข้อมูลได้ทุกครั้งที่ต้องการ การแก้ปัญหาดังกล่าวโดยการผสมผสานระหว่างเทคโนโลยีระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) และระบบจัดการฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Management System) ทำให้สามารถเข้าถึงข้อมูลได้ทุกประเภท โดยอาศัยเทคโนโลยีการจัดการข้อมูลแบบขนาน (Parallel Database Processing) ซึ่งเป็นระบบที่มีประสิทธิภาพสูง โดยข้อมูลทั้งหมดจะถูกรวบรวมไว้ในที่เดียวกันคือ คลังข้อมูลหรือ Data Warehouse

Data Warehouse จึงเป็นคลังข้อมูลขนาดใหญ่ที่เก็บข้อมูลภายในของทั้งองค์กร และข้อมูลภายนอกที่เป็นประโยชน์แก่องค์กร โดยมีการจัดการและบำรุงรักษาอย่างเหมาะสมกับองค์กรนั้นๆ สร้างขึ้นโดยอาศัยการพัฒนาารูปแบบของเทคโนโลยีหลายอย่างผสมผสานกัน ประกอบกับประสิทธิภาพของผู้เชี่ยวชาญของผู้สร้างระบบ สำหรับระบบข้อมูลพื้นฐานที่จัดเก็บไว้ใน Data Warehouse ประกอบด้วยระบบข้อมูล 2 ประเภทคือ Operational Systems และ Informational Systems โดยที่ Operational Systems จะเป็นระบบที่ช่วยในการดำเนินธุรกิจในแต่ละวัน โดยมักจะเน้นที่เรื่องใดเรื่องหนึ่งโดยเฉพาะ เช่น รายการขายสินค้า รายการผลิตสินค้า รายการเกี่ยวกับบัญชี เป็นต้น ในขณะที่ Informational Systems จะเป็นระบบข้อมูลเกี่ยวกับการวิเคราะห์ และตัดสินใจในการดำเนินธุรกิจ ได้แก่ กระบวนการวางแผน ทำนายเหตุการณ์ล่วงหน้า และการบริหาร โดยมักจะครอบคลุมข้อมูลในหลายๆ เรื่อง และอาศัยข้อมูลจาก Operational Systems มาเกี่ยวข้องอย่างมาก ซึ่งทั้งสองระบบนี้ต่างก็มีความสำคัญไม่ยิ่งหย่อนไปกว่ากัน ดังตารางที่ 2.1 เปรียบเทียบความแตกต่างระหว่าง Operational System กับ Data Warehouse

ตารางที่ 2.1 เปรียบเทียบระหว่าง Operational System กับ Data Warehouse

คุณลักษณะ	ฐานข้อมูลประจำวัน	ฐานข้อมูลของคลังข้อมูล
การใช้งาน	ข้อมูล	วิเคราะห์
ลักษณะการจัดการข้อมูล	Application Oriented	Subject Oriented
โครงสร้างของข้อมูล	ซับซ้อนแล้วแต่เครื่องและการคำนวณ แต่จะเป็นรูปแบบชัดเจน ประมวลผลซ้ำเรื่องเดิม	มีโครงสร้างไม่แน่นอน ประมวลผลแบบวิเคราะห์
เนื้อหาและช่วงเวลา	เป็นข้อมูลปัจจุบัน (เพื่อจับความ เป็นไปในปัจจุบัน)	เป็นข้อมูลจากทั้งในอดีตและปัจจุบัน (เพื่อใช้หาความเป็นไป และแนวโน้มในอนาคต)
การอัปเดต	เป็นเรื่อยๆ ไป มีจำนวนน้อยและทำเป็นประจำ	แล้วแต่สถานการณ์และความต้องการ ไม่มีการอัปเดตโดยตรง แต่จะใช้ข้อมูลที่ประมวลผลใหม่ ซ้อนทับขึ้นไปเรื่อยๆ
การเคลื่อนไหวของข้อมูล	ข้อมูลมีการเคลื่อนไหวตลอดเวลา (dynamic)	ข้อมูลคงที่จนกว่าจะปรับปรุงใหม่ (การวิเคราะห์และคำนวณจะทำได้ก็ต่อเมื่อข้อมูลมีปริมาณคงที่เท่านั้น)
เวลาในการทำงาน	เสี้ยววินาทีถึง 2-3 วินาที	ไม่แน่นอน ตั้งแต่หลายวินาทีจนถึงนาที
ความแน่นอนในการใช้ข้อมูล	แน่นอน	ไม่แน่นอน
แหล่งข้อมูล	ภายในองค์กร	ทั้งภายในและภายนอกองค์กร
ขนาดของข้อมูล	Gigabyte	Gigabyte ถึง Terabyte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ความหมายของคลังข้อมูล

บิลล์ อินมอน ให้คำจำกัดความของคลังข้อมูล (Data Warehouse) ไว้ว่า “เป็นแหล่งข้อมูลที่ใช้ในการตัดสินใจของระดับผู้บริหาร ข้อมูลเหล่านี้จะไม่มี การเปลี่ยนแปลง (Nonvolatile) และจะถูกจัดเก็บไว้เป็นเรื่องราว (Subject-Oriented) มีการเก็บลักษณะที่รวบรวม (Integrated) และมีมุมมองในเรื่องเวลาของข้อมูลเป็นสำคัญ (Time Variant)” (Rob and Coronel. 2000 : 579)

### 2.3.1 Integrated

เป็นการรวบรวมข้อมูลจากแหล่งต่างๆเข้าไว้ด้วยกันสร้างขึ้นเป็นฐานข้อมูลที่มีความสอดคล้องกัน ตัวอย่างเช่น ถ้าของตัวแปรตัวเดียวกันในแต่ละฐานข้อมูลอาจจะต่างกัน เช่นฐานข้อมูลหนึ่งอาจใช้ “0” และ “1” แต่ตัวแปรเดียวกันนี้ในอีกฐานข้อมูลหนึ่งใช้ “T” และ “F” เป็นต้น ดังนั้นฐานข้อมูลที่สร้างขึ้นใหม่จะต้องได้รับการกำหนดค่าของตัวแปรให้เหมือนกันเป็นหนึ่งเดียวกัน

### 2.3.2 Subject-Oriented

ข้อมูลที่เกิดขึ้นจะต้องถูกสร้างขึ้นจากหัวข้อเรื่องที่สนใจ ประกอบด้วยหัวข้อที่เก็บเฉพาะข้อมูลที่เป็นต่อกระบวนการตัดสินใจเท่านั้น เช่น ถ้าบริษัทประกันภัยต้องการสร้าง Data Warehouse ฐานข้อมูลที่ได้จะต้องสร้างขึ้นจากประวัติลูกค้า (Customer) เบี้ยประกัน (Premium) และการเรียกร้อง (Claim) เป็นต้น แทนที่จะแยกเก็บตามประเภทของการประกันภัยเป็นประกันชีวิต, ประกันรถยนต์, ประกันสุขภาพและประกันอุบัติเหตุ เป็นต้น

### 2.3.3 Time Variant

ข้อมูลที่เกิดขึ้นจะมีความสัมพันธ์กับเวลาเป็นสำคัญ เนื่องจากในการวิเคราะห์และตัดสินใจจำเป็นต้องมีข้อมูลเปรียบเทียบในแต่ละช่วงเวลาที่แตกต่างกัน ข้อมูลที่เก็บใน Data Warehouse จึงต้องมีเรื่องของเวลาเข้ามาเกี่ยวข้องเสมอทำให้สามารถเกิดการเปรียบเทียบข้อมูลตามช่วงเวลาต่างๆ ได้เช่น เปรียบเทียบตามรายปี รายไตรมาส รายเดือน หรือรายสัปดาห์ เป็นต้น หากข้อมูลที่จัดเก็บไม่ได้มีเวลาเข้ามาเกี่ยวข้องก็จะไม่สามารถเปรียบเทียบข้อมูลต่างๆ เหล่านี้ได้เลย นอกจากนั้นข้อมูลที่เก็บก็เพื่อจะนำมาใช้ในการวิเคราะห์และตัดสินใจ จึงควรจะต้องมีช่วงเวลาการเก็บข้อมูลย้อนหลังค่อนข้างนานบางทีข้อมูลอาจมีอายุประมาณ 5-10 ปี เพื่อการเปรียบเทียบ หาแนวโน้มและทำนายผลลัพธ์ในอนาคตข้อมูลได้อย่างถูกต้องแม่นยำยิ่งขึ้น

### 2.3.4 Non-volatile

เนื่องจากข้อมูลที่เก็บใน Data Warehouse เป็นการเก็บข้อมูลย้อนหลังหรือเป็นข้อมูลในอดีต ดังนั้นข้อมูลต่างๆ ที่จัดเก็บไว้แล้วจะเป็นข้อมูลที่คงที่ ไม่เปลี่ยนแปลง เป็นข้อมูลประเภทที่สามารถอ่านได้เพียงอย่างเดียว (Read-Only) ทั้งนี้ก็เพื่อเป็นการรักษาความถูกต้องและความสอดคล้องกัน ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับข้อมูลที่ใช้ในงานประจำวัน ผู้ใช้ไม่สามารถเข้าไปทำการเปลี่ยนแปลงแก้ไขข้อมูลใดๆ ภายในฐานข้อมูลได้ ทำได้เพียงการโหลดและเข้าถึงข้อมูลเท่านั้น ซึ่งการเรียกค้นข้อมูลทำได้โดยการใช้เครื่องมือวิเคราะห์ข้อมูลประเภทต่างๆ

## 2.4 จุดประสงค์ของคลังข้อมูล

คลังข้อมูลจะเก็บรักษาค่าของข้อมูลที่มีความเสถียรและได้รับการตรวจสอบแล้ว ซึ่งอาจมีประโยชน์ในการนำไปเปรียบเทียบระหว่างคลังข้อมูลกับฐานข้อมูลประจำวัน ดังต่อไปนี้ (พรพิมล อนันควณิช, 2544 : 5-6)

- ฐานข้อมูลประจำวันจะช่วยให้เราปฏิบัติกิจกรรมต่างๆ ได้สำเร็จ และคลังข้อมูลจะช่วยในการตัดสินใจ ตัวอย่างเช่น ฐานข้อมูลประจำวันอาจจะชี้ให้เห็นว่ามีที่นั่งใดว่างบ้างในเที่ยวบินหนึ่งๆ เพื่อที่เจ้าหน้าที่ของบริษัททัวร์จะได้สำรองที่นั่งใหม่ได้ ในทางตรงกันข้ามในคลังข้อมูลอาจแสดงให้เห็นถึงลักษณะรูปแบบของที่นั่งที่ว่างในแต่ละเที่ยวบินที่ผ่านๆ มา เพื่อผู้จัดการสายการบินจะตัดสินใจได้ว่าจะต้องเปลี่ยนแปลงเที่ยวบินในอนาคตหรือไม่

- ฐานข้อมูลประจำวันจะมีการเปลี่ยนแปลงที่รวดเร็วและในทันทีทันใด คือข้อมูลจะมีการเปลี่ยนแปลงอยู่เสมอ เช่น เมื่อมีรายการการสั่งซื้อใหม่เข้ามา หรือการสั่งซื้อถูกยกเลิก หรือเมื่อมีการผลิตหรือขนส่งผลิตภัณฑ์ หรือมีการสำรองที่นั่งใหม่ ส่วนคลังข้อมูลจะมีเสถียรภาพ คือข้อมูลจะถูกปรับปรุงเป็นระยะๆ ตามมาตรฐานที่กำหนดไว้ ซึ่งอาจเป็นทุกเดือน ทุกอาทิตย์ หรือแม้แต่ทุกชั่วโมง ซึ่งการปรับปรุงจะเพิ่มค่าให้แก่ช่วงเวลาใหม่เท่านั้น จะไม่มีการเปลี่ยนแปลงข้อมูลที่เคยไว้ในคลังข้อมูลก่อนหน้านั้นเลย

- ฐานข้อมูลประจำวันมุ่งให้ความสนใจที่รายละเอียด คือในการสำรองที่นั่งนั้น เจ้าหน้าที่ของบริษัททัวร์ไม่ต้องการที่จะรู้จำนวนเฉลี่ยของที่นั่งว่างในแต่ละเที่ยวบิน แต่คลังข้อมูลจะมุ่งเน้นที่การรวมค่าระดับสูงคือผู้จัดการสายการบินที่ทำการปรับปรุงตารางการบินคงไม่สนใจว่ามีที่นั่งว่างที่ไหนบ้างแต่ต้องการเห็นภาพรวม ความหมายของความแตกต่างข้อนี้ก็คือ ค่าที่สำคัญหรือค่าหลักในคลังข้อมูลจะต้องมีค่าเป็นตัวเลขที่สามารถสรุปได้

- โดยทั่วๆ ไปฐานข้อมูลประจำวันจะให้ค่าของข้อมูลที่จะถูกเก็บไว้ในคลังข้อมูลต่อไป

## 2.5 เทคโนโลยีที่สนับสนุนคลังข้อมูล

ปัจจุบันด้วยการพัฒนาและเติบโตขึ้นของเทคโนโลยีต่างๆ มากมายอย่างรวดเร็ว มีส่วนสนับสนุนให้ Data Warehouse เป็นที่ยอมรับกันอย่างแพร่หลาย เทคโนโลยีต่างๆ เหล่านี้ได้แก่ (อนุชิต สุเมธวิทย์, 2541 : 143)

เอกสารนี้เป็นเอกสารที่สงวนสิทธิ์ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เทคโนโลยีของการประมวลผลแบบไคลเอนต์/เซิร์ฟเวอร์ และการพัฒนาทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่มีประสิทธิภาพและทำงานได้ซับซ้อนมากขึ้น พร้อมกับสถาปัตยกรรมเมโมรีที่เหนือชั้น โปรเซสเซอร์ความเร็วสูง และแหล่งเก็บข้อมูลความจุสูง ในเวลาเดียวกันระบบจัดการฐานข้อมูลหรือ DBMS สมัยใหม่รองรับโครงสร้างข้อมูลที่ซับซ้อนได้ดีขึ้นและสนับสนุนมาตรฐานมิดเดิลแวร์ด้วยฮาร์ดแวร์และซอฟต์แวร์เหล่านี้

- เทคโนโลยีทางการวิเคราะห์ข้อมูล ด้วยความได้เปรียบของ OLAP (Online Analytical Processing) ที่มีเหนือกว่าการวิเคราะห์ข้อมูลทางการตลาด และการเงินที่ใช้กันอยู่ทั่วไป ด้วย OLAP ทำให้องค์กรสามารถค้นหาข้อมูลลูกค้า ข้อมูลผลิตภัณฑ์ ข้อมูลการปฏิบัติงาน ตลอดจนกิจกรรมทางธุรกิจ ที่เก็บอยู่ในฐานข้อมูลขนาดใหญ่ได้อย่างสะดวกง่ายดาย ทำให้การตัดสินใจในเชิงกลยุทธ์ทำได้รวดเร็วทันทั่วถึง ตลอดจนช่วยลดค่าใช้จ่ายในการปฏิบัติงาน และช่วยเพิ่มรายได้

- เทคโนโลยีทางด้านอินเทอร์เน็ต ก็ทำให้ Data Warehouse น่าสนใจขึ้นเช่นกัน เพราะผู้ใช้ส่วนใหญ่ชอบอะไรที่ง่ายๆ ต้องการใช้งานโปรแกรมที่ไม่ซับซ้อน เว็บจึงเป็นคำตอบที่น่าสนใจที่สุด ทั้งเรื่องของราคาโดยรวมที่ถูกและใช้งานง่าย

- เทคโนโลยีทางด้านเอเจนต์ (Agent Technology) หากผู้ใช้มีความต้องการวิเคราะห์ข้อมูลบางอย่างทุกวันและต้องการให้มีรายงานทันทีที่พบสิ่งผิดปกติ เอเจนต์จะเข้ามารองรับความต้องการด้านนี้ โดยผู้ใช้สามารถตั้งเวลาให้เอเจนต์ทำงานตามตารางเวลาได้ และกำหนดให้เตือนหากพบสิ่งผิดปกติตามเงื่อนไขที่ตั้งไว้ได้

## 2.6 Data Mart กับ Data Warehouse

ด้วยความแตกต่างระหว่าง Data Mart กับ Data Warehouse ดังตารางที่ 2.2 ที่ว่าข้อมูลใน Data Mart เป็นข้อมูลระดับแผนก ซึ่งอาจจะเป็นส่วนหนึ่งของ Data Warehouse หรืออาจไม่เกี่ยวกันเลยก็ได้ ขนาดของ Data Mart ก็ไม่จำเป็นว่าต้องเล็กกว่าขนาดของ Data Warehouse เสมอไป อยู่ที่ว่าแผนกนั้นต้องการให้ Data Mart เก็บข้อมูลย้อนหลังมากขนาดไหนด้วย

ตารางที่ 2.2 เปรียบเทียบระหว่าง Data Mart กับ Data Warehouse

Feature	Data Mart	Data Warehouse
Size	Averages 50 GB; possible as large as 250 GB	Often terabyte +
Purpose	Subject-area-specific	Enterprise repository
How initiated	Bottom-up	Top-down, along with formed enterprise data model
Control	Departmental	IS
Time line	3 to 6 months	1 to 2 years
Cost	\$10,000 to \$1,000,000	Millions of dollars

ในการวางแผน Data Warehouse ก่อนอื่นต้องมีการตัดสินใจว่าจะเลือกใช้แนวทางไหน จะเริ่มที่ Data Warehouse ก่อน หรือ Data Mart ก่อน มีนักวิจัยบางกลุ่มกล่าวว่า Data Mart เป็นส่วนหนึ่งของ Data Warehouse และจะต้องสร้าง Data Warehouse ก่อนถึงจะสร้าง Data Mart ได้ บ้างก็ว่า Data Warehouse ต้องใช้เวลานานเป็นปีหรือหลายๆ ปี เห็นผลช้าและเสี่ยงต่อการลงทุนสูง ก็น่าที่จะสร้าง Data Mart ก่อนโดยเน้นเฉพาะแผนกที่สำคัญทำกำไรให้กับบริษัทเป็นจำนวนมาก ก็น่าจะได้ผลลัพธ์เร็วกว่า ซึ่งการที่เลือกใช้แนวทางไหนจะเริ่มที่ Data Warehouse ก่อนหรือ Data Mart ก่อน ก็ขึ้นอยู่กับองค์กรมีความพร้อมขนาดไหนและขึ้นกับลักษณะขององค์กรว่าเป็นอย่างไรด้วย ผู้พัฒนาต้องพิจารณาเลือกใช้ให้เหมาะสมกับความต้องการและโครงสร้างของแต่ละองค์กร

วิธีการเริ่มต้นที่ Data Warehouse ก่อนเรียกอีกอย่างว่าการเริ่มจากบนลงล่าง (Top Down Approach) เพราะเป็นการเริ่มต้นจากความต้องการระดับองค์กร และนำไปสู่การใช้เทคโนโลยีสร้าง Data Warehouse ให้มาเข้ากับความต้องการขององค์กร ถ้าแผนกไหนมีความต้องการใช้ข้อมูลมากกว่าที่มีอยู่ใน Data Warehouse หรือต้องการจัดการข้อมูลด้วยตนเองไม่ขึ้นกับใคร ก็ต้องดึงข้อมูลจาก Data Warehouse ไปสร้างเป็น Data Mart ในเครื่องของแต่ละแผนก ข้อดีของวิธีนี้คือ คนพัฒนา Data Warehouse จะไม่หลงทางเพราะมองเห็นเป้าหมายชัดเจน สามารถกำหนดจุดสนใจและขอบเขตของ Data Warehouse ลงไปได้เลย แต่ข้อจำกัดของวิธีนี้ก็คือ เราต้องรู้ความต้องการขององค์กร แล้วว่าขอบเขตของงาน Data Warehouse มีแค่ไหน ผู้บริหารในองค์กรมองเห็นภาพว่าจะนำ Data Warehouse มาใช้อย่างไร ซึ่งในความเป็นจริงแล้วเป็นไปได้ยากมาก เพราะแม้แต่ของเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถของเทคโนโลยี Data Warehouse เองก็ขึ้นอยู่กับเครื่องมือที่จะมาครอบมันอีกที อีกทั้งทางผู้บริหารเองเกือบทั้งหมดจะยังมองไม่เห็นภาพตั้งแต่แรกอยู่แล้ว

ขณะที่วิธีที่เริ่มจาก Data Mart ก่อนเรียกอีกอย่างว่า การเริ่มจากล่างขึ้นบน (Bottom Up Approach) วิธีนี้จะเริ่มจากการสร้างระบบทดลองขึ้นมา โดยอาศัยจากเทคโนโลยีที่มีอยู่ วิธีนี้จะต้องเข้าใจความต้องการแค่บางส่วนขององค์กร ผลงานจะออกมาได้เร็วเพราะเกี่ยวข้องกับคนกลุ่มเล็ก และมองความต้องการระดับแผนกแทนที่จะเป็นขององค์กรทั้งหมด ถ้าระบบทดลองไม่สำเร็จก็ไม่เสียหายมากนัก เพราะค่าใช้จ่ายค่อนข้างต่ำ แต่ข้อเสียของวิธีนี้ก็คือ เมื่อพัฒนาไปถึงจุดหนึ่งที่เราสร้าง Data Mart ขึ้นมาสำเร็จแล้ว มักจะมีปัญหาเมื่อจะค่อยๆ ขยับระดับงานขึ้นมามองในระดับองค์กรเพื่อทำเป็น Data Warehouse จะเป็นการดีที่เราจะมีเวลาที่ขย้อนกลับขึ้นมาดูภาพรวมขององค์กร เพื่อที่ว่าเราจะได้วางแผนตั้งแต่แรกที่จะสร้าง Data Warehouse สำหรับขององค์กรต่อไป (อนุชิต สุเมธวิทช์. 2541 : 145)

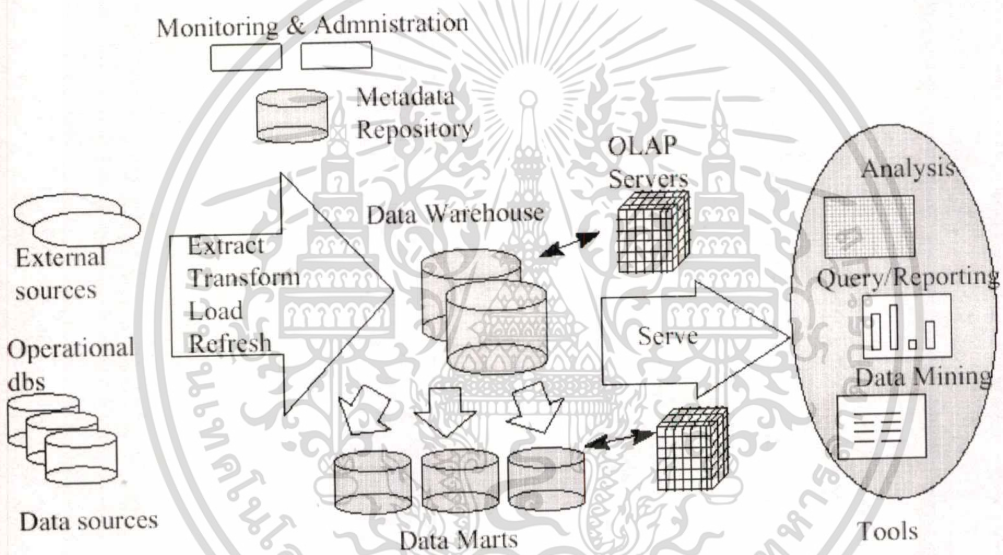


## บทที่ 3

### สถาปัตยกรรมของคลังข้อมูล

#### 3.1 สถาปัตยกรรมของคลังข้อมูล

สถาปัตยกรรมของคลังข้อมูลประกอบด้วยส่วนประกอบต่างๆ ดังรูปที่ 3.1



รูปที่ 3.1 Data Warehousing Architecture

#### 3.1.1 ที่มาของข้อมูล (Data Source)

ที่มาของข้อมูลซึ่งอาจได้มาจากแหล่งข้อมูลภายในองค์กร หรือจากแหล่งข้อมูลภายนอกอื่นๆ ที่เกี่ยวข้องด้วยก็ได้ ซึ่งข้อมูลภายในองค์กรได้มาจากฐานข้อมูลประเภทต่างๆ ที่อยู่บนเครื่อง Host โดยเป็นข้อมูลจากส่วนปฏิบัติงานประจำวันซึ่งมีลักษณะการประมวลผลแบบ Online Transaction Processing (OLTP) หรือเป็นข้อมูลที่ผ่านการประมวลผลแบบ batch ก็ได้

### 3.1.2 การดึงข้อมูล การแปลงข้อมูลและการโหลดข้อมูล (Data Extraction , Data Transformation and Loading)

เป็นกระบวนการดึงข้อมูล ทำความสะอาดข้อมูล ตรวจสอบความถูกต้องของข้อมูล แปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมต่อการนำไปใช้งาน ตลอดจนการโหลดข้อมูลที่ผ่านขั้นตอนดังกล่าวไปไว้บน Data Warehouse

- การทำความสะอาดข้อมูล (Data Cleansing) หรือการสะสางข้อมูล (Cleanup Data) หมายถึงการปรับปรุงคุณภาพของข้อมูลให้ผิดน้อยที่สุด นับเป็นขั้นตอนที่มักใช้เวลานานที่สุด เนื่องจากความจริงที่ว่าข้อมูลขององค์กรส่วนใหญ่แล้วยังมีส่วนที่ผิดอยู่ ซึ่งนอกจากจะมีการดึงข้อมูลมาจากหลายๆ แหล่งมาไว้ด้วยกันแล้ว บางครั้งยังพบว่าการดึงข้อมูลชุดเดียวกันจากคนละแหล่ง ก็ยังได้ข้อมูลที่ไม่ตรงกันอีกด้วย ทำให้การทำความสะอาดข้อมูลจึงเป็นงานที่ต้องทำซ้ำๆ หลายๆ ครั้งเพื่อให้ได้ข้อมูลที่ถูกต้องมากที่สุด ในการทำความสะอาดข้อมูลจึงต้องมีการตัดสินใจว่าจะต้องทำให้ข้อมูลถูกต้องขนาดไหนถึงจะยอมรับได้ คำตอบก็ขึ้นอยู่กับว่าเราจะเอา Data Warehouse นั้นไปทำอะไร

- การแปลงข้อมูล (Transformation) เกิดมาจากการที่ข้อมูลในแต่ละเครื่องเก็บข้อมูลคนละฟอร์แมต แต่ข้อมูลบางอย่างหมายถึงสิ่งเดียวกัน Data Warehouse จำเป็นต้องเก็บข้อมูลเหล่านั้นในรูปแบบที่เป็นมาตรฐานเดียวกัน โดยจะต้องเลือกเก็บแบบใดแบบหนึ่ง หรือเลือกเก็บแบบใหม่อีกแบบก็ได้

- การซั้มาไลเซชัน (Summarization) และการแอกกรีเกต (Aggregation) โดยการซั้มาไลเซชันก็คือการรวมข้อมูลหรือการสรุปข้อมูล และการแอกกรีเกตคือการเก็บข้อมูลที่รวมนั้นไว้ในฐานข้อมูล ซึ่งการรวมข้อมูลนั้นก็ใช้ว่าจะทำได้ง่าย ๆ มีเรื่องที่ต้องคิดมากกว่านั้น เช่น การรวมจำนวนลูกค้ารายเดือนแล้วเก็บไว้ จะรู้ได้อย่างไรว่า รายการซื้อขายสินค้าในแต่ละทรานแซกชัน รายการไหนเป็นลูกค้าคนเดียวกันบ้าง หากนำข้อมูลทั้งหมดมารวมกันเลยก็จะได้จำนวนลูกค้าเกินจริง หรือการรวมจำนวนยอดขายตามประเภทสินค้ารายสามเดือน ถ้าสินค้าบางอย่างจัดอยู่ในประเภทสินค้าได้มากกว่าหนึ่งประเภท การรวมข้อมูลสินค้าทุกประเภทเข้าด้วยกันเลยก็จะให้ข้อมูลที่ไม่วถูกต้อง หรืองานบางอย่างจะนำข้อมูลมารวมกันเฉยๆ ไม่ได้ อาจต้องหาค่าเฉลี่ยก่อนแล้วค่อยเก็บไว้ก็มี เป็นต้น

- การโหลดข้อมูล (Loading) ก็เป็นอีกงานที่ต้องทำซ้ำๆ เหมือนกันและยังต้องคอยตรวจสอบคุณภาพของข้อมูลที่ต่อเครื่องและระบบเน็ตเวิร์กด้วย เพราะงานโหลดข้อมูลจะกินซีพียูใหม่ของเครื่องในการคัดลอกข้อมูลออกไปยัง Data Warehouse และการพยายามที่จะรีเฟรชข้อมูลจากแหล่งข้อมูลให้ตรงกันกับข้อมูลใน Data Warehouse ตลอดเวลา ก็เป็นการใช้งานเน็ตเวิร์กอย่างหนักเลยทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ สิ่งที่ต้องคำนึงถึงอีกอย่างในการโหลดข้อมูล ก็คือการวางแผนการอัปเดตเครื่อง และฮาร์ดดิสก์ไว้ล่วงหน้าตั้งแต่แรก เนื่องจากข้อมูลใน Data Warehouse นั้นมีการขยายขนาดเร็วมากเพราะมีลักษณะการเก็บข้อมูลเป็นรายวัน รายสัปดาห์ รายเดือน บางทีต้องมีการเก็บข้อมูลบางส่วนขึ้นเทปบ้าง จึงควรต้องมีการวางแผนและตัดสินใจว่าจะเก็บข้อมูลใน Data Warehouse ใ้ไวมากน้อยหรือนานเพียงใด

การเลือกเครื่องมือที่ใช้ในการทำความสะอาดข้อมูล หรือเครื่องมือที่ใช้ในการกำจัดข้อมูลขยะที่ก่อปัญหาให้กับ Data Warehouse หรือเป็นการกำจัดข้อมูลที่ไร้ค่าออกจากข้อมูลที่ใช้ในการตัดสินใจเชิงกลยุทธ์ อาจทำได้โดยการเขียนสคริปต์ขึ้นใช้เอง (Custom scrubbing scripts) หรือใช้เครื่องมือพิเศษที่ใช้ในการดึง (Extraction) แปลง (Transformation) และโหลด (Load) ข้อมูลที่เรียกว่า ETL เพื่อช่วยให้ทำงานได้สะดวกรวดเร็วและปลอดภัยยิ่งขึ้นก็ได้

### 3.1.3 Data Modeling Tools

เป็นการนำเครื่องมือที่ใช้ช่วยในการกำหนดแบบจำลองของข้อมูล เพื่อให้การรวบรวมข้อมูลจากแหล่งข้อมูลหลายๆ แห่งเข้าสู่ฐานข้อมูลปลายทางเป็นไปอย่างถูกต้อง

### 3.1.4 Target Database

เป็นฐานข้อมูลปลายทางที่ใช้เป็น Data Warehouse สำหรับจัดเก็บข้อมูลที่ดึงมาจากแหล่งข้อมูลต่างๆ ซึ่งผ่านการปรับปรุงให้อยู่ในรูปแบบที่ถูกต้องเหมาะสม สำหรับไว้เรียกใช้งานต่อไป โดยเครื่องมือที่ใช้วิเคราะห์ข้อมูล

### 3.1.5 Metadata Repository

เป็นแหล่งที่ใช้เก็บรายละเอียดข้อมูลของข้อมูล โดย Metadata จะทำหน้าที่เป็นเหมือนอินเด็กซ์และสารบัญของหนังสือ ที่ช่วยให้ผู้ใช้รู้ว่าเนื้อหาอะไรบ้างและจะหาได้จากที่ไหน ซึ่งก็ทำให้ผู้ใช้รู้ว่ามียอะไรอยู่ใน Data Warehouse บ้างและจะมาใช้ได้อย่างไรจากที่ไหน ตัวอย่างรายละเอียดต่างๆ ที่ Metadata ต้องเก็บ ได้แก่

- เก็บข้อมูลเกี่ยวกับพจนานุกรมข้อมูล
- เก็บ Directory of content ซึ่งจะเป็นตัวบอกว่ามีอะไรอยู่ใน Data Warehouse และอยู่ที่ไหน นอกจากนี้ต้องมีคำอธิบายของข้อมูลแต่ละส่วน ทั้งคำอธิบายสำหรับทางผู้พัฒนาและผู้ใช้ด้วย
- กฎเกณฑ์และเงื่อนไขในการที่จะค้นหาข้อมูล หรือกฎในการ Drill-down, Roll-up ของโดเมนชั้นที่มีลักษณะเป็นไฮราติ เช่น ข้อมูลวัน → สัปดาห์ → เดือน → 3 เดือน → ปี เป็นต้น
- ระยะเวลาการลบข้อมูลใน Data Warehouse ทุกๆ กี่ปี หรือใช้อะไรในการตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทุกอย่างที่เกิดขึ้นระหว่างการนำข้อมูลเข้าจากแหล่งข้อมูลต้นทางไปจนกระทั่งถึงการนำข้อมูลไปเก็บไว้ใน Data Warehouse เช่น ข้อมูลจากแหล่งข้อมูลที่เกี่ยวข้องมีอะไรบ้างและอยู่ที่ไหน ความถี่ในการดึงข้อมูล เทเบิลอะไรในฐานข้อมูลต้นทางไปเป็นเทเบิลอะไรใน Data Warehouse อย่างไร มีการแปลงข้อมูลจากอะไรไปเป็นอะไร เช็กความถูกต้องของข้อมูลอย่างไร อัลกอริทึมในการ ضمมาไลเซชันและเอคกริเกชัน เป็นต้น

ประโยชน์ของ Metadata สำหรับผู้พัฒนา Data Warehouse คือมันจะทำหน้าที่เก็บการเปลี่ยนแปลงที่เกิดขึ้นใน Data Warehouse เนื่องมาจากการเปลี่ยนแปลงที่ตัวข้อมูลต้นทางเอง หรือจากการเปลี่ยนแปลงโครงสร้างขององค์กรเองด้วย นอกจากนี้ในส่วน Metadata ที่เก็บขั้นตอนการดึงและแปลงข้อมูล ก็จะทำให้สามารถทำงานในส่วนนี้ซ้ำๆ ได้โดยใช้ขั้นตอนเดิม ซอฟต์แวร์บางตัวที่จัดการเรื่องการดึงข้อมูล เช่น Prism Data Warehouse หรือ Carleton Passport ก็จะเก็บข้อมูลส่วนที่เกี่ยวข้องกับการดึงข้อมูลไว้ใน Metadata และใช้ข้อมูลใน Metadata ในขั้นตอนการดึงข้อมูลของมัน ในความเป็นจริงแล้วผู้พัฒนา ก็สามารถสร้าง Metadata ขึ้นเองได้คือเก็บข้อมูล Metadata ไว้ในรูปเทเบิลในฐานข้อมูล แต่อย่างไรก็ดี ซอฟต์แวร์บางตัวที่เลือกใช้ในแต่ละงานก็มี Metadata ของมันเองเช่นกัน ซึ่ง Metadata ของซอฟต์แวร์นั้นๆ จะเก็บอะไรบ้างก็ขึ้นอยู่กับว่าซอฟต์แวร์นั้นทำงานส่วนไหนใน Data Warehouse

ปัจจุบันยังขาดรูปแบบของ Metadata ที่เป็นมาตรฐานที่ยอมรับกัน ก็เลยกลายเป็นว่าต่างคนต่างสร้างฟอร์มเมตของตัวเอง อย่งไรก็ตาม ในปัจจุบันได้มีความพยายามที่จะสร้างมาตรฐานของ Metadata ขึ้นมาซึ่งประโยชน์ที่จะได้รับการมีมาตรฐานคือ สามารถรวม Data Mart ของหลายๆ แผนกที่ใช้คนละแพลตฟอร์มเข้าด้วยกันได้ หรือรวม Data Warehouse ของสองบริษัทที่ควบกิจการเข้าด้วยกันก็ได้ นอกจากนี้ยังสามารถใช้ข้อมูลร่วมกันระหว่างบริษัทโดยผ่านอินเทอร์เน็ตก็ได้

### 3.1.6 Data Access and Analysis Tools

เป็นการเข้าถึงข้อมูลหรือการนำข้อมูลที่เกิดขึ้นอยู่ใน Data Warehouse ออกมาใช้โดยเครื่องมือวิเคราะห์ข้อมูล อาจกล่าวได้ว่าหากปราศจากเครื่องมือสำหรับเอนด์ยูสเซอร์ที่เหมาะสมสำหรับเข้าถึงข้อมูลและวิเคราะห์ข้อมูลแล้ว การอิมพลิเมนต์ Data Warehouse ก็จะไม่มีความหมายโดยสิ้นเชิง ซึ่งปัจจุบันก็มีเครื่องมืออยู่หลายประเภทให้เลือกใช้ตามความเหมาะสม ได้แก่

- Querying and Reporting tools
- OLAP tools
- Executive Information Systems (EIS)
- Data Mining applications

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 สถาปัตยกรรมอื่นที่เกี่ยวข้องในการสร้าง Data Warehouse

ความสำเร็จของการสร้าง Data Warehouse อยู่ที่การเลือกสถาปัตยกรรมสำคัญ 3 ประการอย่างถูกต้องเพื่อให้ประสานการทำงานกันอย่างเหมาะสมในการรวมกันเป็นระบบ Data Warehouse ดังนี้ (Weldon and Joch. 2540 : 104-105)

3.2.1 เลือกสถาปัตยกรรมของโครงสร้างข้อมูล ซึ่งเป็นสถาปัตยกรรมโดยรวมก่อน การตัดสินใจในจุดนี้จะส่งผลกระทบต่อคอมพิวเตอร์ที่จะเลือกใช้ในอนาคตอย่างมากด้วย

- Federated Architecture
- Tiered Architecture

#### 3.2.2 เลือกสถาปัตยกรรมของเซิร์ฟเวอร์

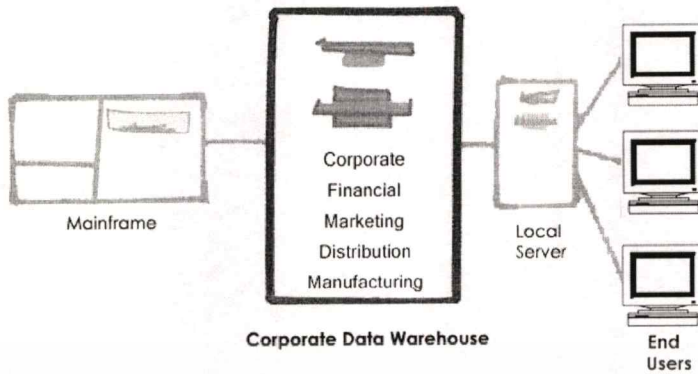
- Single Processing
- Symmetric Multiprocessing (SMP)
- Massive Parallel Processing (MPP)

#### 3.2.3 เลือกสถาปัตยกรรมของระบบจัดการฐานข้อมูล

- Relational Database
- Multidimensional Database
- Object-relational Database

### 3.3 สถาปัตยกรรมของโครงสร้างข้อมูล

เมื่อเริ่มจะพัฒนา Data Warehouse ควรจะเริ่มพิจารณาจากโครงสร้างของฐานข้อมูลของ Data Warehouse ทั้งด้าน Logical และ Physical โดยอาจเลือกใช้แพลตฟอร์มแบบรวมศูนย์ (Centralized) ที่ติดตั้งอยู่ในบริเวณเดียวกัน ดังรูปที่ 3.2 หรือแบบกระจาย (Distributed) ที่แยกฐานข้อมูลออกไปตามสำนักงาน จังหวัดหรือประเทศต่างๆ ก็ได้ โดยในบางบริษัทอาจรวบรวมข้อมูลจากแหล่งข้อมูลภายในและภายนอกหลายแหล่งมาเก็บไว้ใน Data Warehouse เดียวกัน แนวทางรวบรวมข้อมูลไว้ด้วยกันเช่นนี้ ช่วยเพื่อพลังในการประมวลผลข้อมูลที่มีอยู่

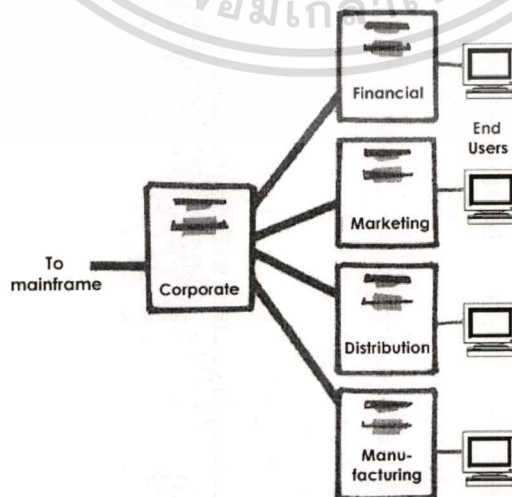


รูปที่ 3.2 สถาปัตยกรรม Data Warehouse แบบรวมศูนย์

ในสถาปัตยกรรมแบบรวมศูนย์ Data Warehouse จะรวมข้อมูลเป็นหนึ่งเดียวตอบสนองธุรกิจในทุกรูปแบบ ข้อมูลแต่ละหัวเรื่องที่แยกกันอยู่บนฐานข้อมูลจะมีความสัมพันธ์กัน และจะถูกเก็บจริงไว้ในแพลตฟอร์มเดียวกัน ให้ประสิทธิภาพด้านการประมวลผลและลดค่าใช้จ่ายลง ซึ่งมีสองทางเลือกสำหรับสถาปัตยกรรมแบบรวมศูนย์ที่ต้องพิจารณาคือ (Weldon, 2540 : 124-125)

### 3.3.1 Federated Architecture

มีการกระจายข้อมูลออกไปยังฐานข้อมูลหลายตัวตามฟังก์ชัน ดังนั้นด้วยสถาปัตยกรรมเช่นนี้ ข้อมูลด้านการเงินอาจถูกเก็บอยู่ในเซิร์ฟเวอร์ของไซต์แห่งหนึ่ง ขณะที่ข้อมูลด้านการตลาดและข้อมูลด้านการผลิตอาจถูกแยกออกไปยังอีกแห่งหนึ่ง ดังรูปที่ 3.3 โดยจะเป็นการรวมข้อมูลเข้าด้วยกันทาง logical แต่ข้อมูลจริงแยกกันอยู่บนฐานข้อมูลหลายตัวที่เชื่อมต่อกันทาง physical ซึ่งอาจอยู่บนไซต์เดียวกัน หรือต่างไซต์ก็ได้

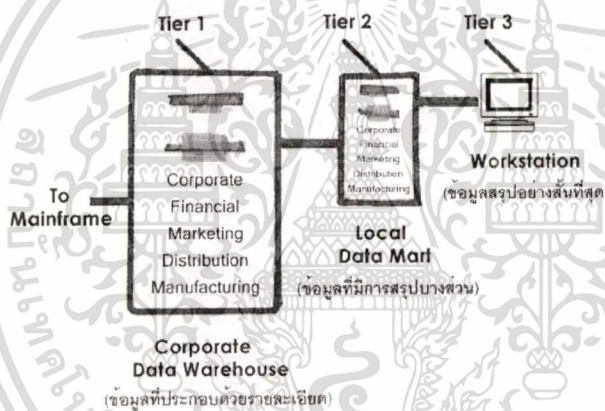


รูปที่ 3.3 Federated Architecture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 Tiered Architecture หรือสถาปัตยกรรมแบบลำดับขั้น

สถาปัตยกรรมแบบนี้จะเก็บข้อมูลสรุปที่มีข้อมูลกระชั้นที่สุดบน Workstation ของผู้ใช้ในระดับแรก ขณะที่ข้อมูลบนเซิร์ฟเวอร์ในระดับที่สองจะมีรายละเอียดเพิ่มขึ้น และมีรายละเอียดข้อมูลสูงที่สุดในระดับที่สาม ดังรูปที่ 3.4 หรืออาจกล่าวได้ว่าการจัดเก็บข้อมูลจะถูกจัดเก็บโดยแบ่งตามระดับความละเอียดของข้อมูล โดยเซิร์ฟเวอร์ในระดับแรก (Tier 1) จะจัดการกับคำสั่งเรียกค้นข้อมูลทั้งหมด จะเหลือคำสั่งเรียกค้นจำนวนไม่มากนักที่จะผ่านไปยังระดับที่สอง (Tier 2) และสาม (Tier 3) เพื่อค้นรายละเอียดที่มากขึ้น คุณลักษณะของคอมพิวเตอร์ที่ระดับที่หนึ่งจะสามารถรองรับโหลดการใช้งานได้สูง เนื่องจากมีปริมาณข้อมูลที่ไม่มากนัก ขณะที่เซิร์ฟเวอร์ในระดับสองและสามจะเพิ่มประสิทธิภาพในการประมวลผลข้อมูลปริมาณมาก แต่จะรับโหลดได้ไม่มากนัก



รูปที่ 3.4 Tiered Architecture

### 3.4 สถาปัตยกรรมของเซิร์ฟเวอร์

นอกจากการตัดสินใจเลือกใช้สถาปัตยกรรมรวมของ Data Warehouse แบบรวมศูนย์ หรือแบบกระจายแล้ว สิ่งที่ต้องพิจารณาประกอบกันไปคือเซิร์ฟเวอร์ที่จะเก็บและส่งผ่านข้อมูล เริ่มจากติดตั้งเซิร์ฟเวอร์ขนาดเท่าใด บริษัทที่มีการขยายระบบไปอีกมากน้อยแค่ไหน มีระบบดังกล่าวให้ใช้แล้วหรือไม่ และจัดการดูแลระบบได้อย่างไร ทั้งหมดนี้มีอิทธิพลต่อการเลือกสถาปัตยกรรมของเซิร์ฟเวอร์ทั้งสิ้น ดังรูปที่ 3.5 (Weldon, 2540 : 125-126)

#### 3.4.1 Single Processing Server

หรือเซิร์ฟเวอร์แบบที่มีโพรเซสเซอร์เดียว มีข้อดีที่จัดการได้ง่าย แต่ก็มีข้อด้อยที่ข้อจำกัดด้านพลังประมวลผลและความสามารถในการขยายระบบ นอกจากนั้นซึ่งเกิดเซิร์ฟเวอร์นั้นมี Single point of failure ทำให้มีความเสี่ยงสูง Data Warehouse ที่ทำงานบนเซิร์ฟเวอร์แบบนี้จึงมีข้อจำกัด  
เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

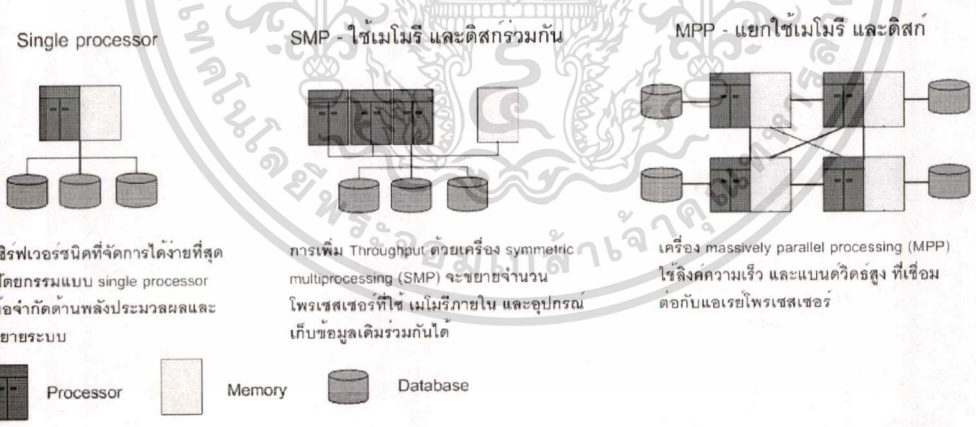
ด้านเสถียรภาพ หากจะขยายซึ่งเกิดเซิร์ฟเวอร์เน็ตเวิร์กผ่านสถาปัตยกรรมแบบกระจายโดยการใช้  
 มิดเดิลแวร์เพื่อกระจาย traffic ข้ามเซิร์ฟเวอร์หลายตัว สถาปัตยกรรมเหล่านี้เพิ่มคุณสมบัติด้าน  
 เสถียรภาพขึ้น เนื่องจากสามารถย้ายการทำงานไปยัง Backup เซิร์ฟเวอร์ ในกรณีถ้าเซิร์ฟเวอร์ใด  
 เซิร์ฟเวอร์หนึ่งล้มเหลวขึ้นมา แต่การจัดการระบบจะซับซ้อนยิ่งขึ้นมาก

**3.4.2 Symmetric Multiprocessing Server (SMP)**

สามารถเพิ่ม Throughput ได้โดยเพิ่มโพรเซสเซอร์เข้าไป แต่ยังใช้หน่วยความจำภายในและ  
 อุปกรณ์เก็บข้อมูลเดียวกัน การขยายระบบของเครื่อง SMP จะมีขีดจำกัดที่จำนวนโพรเซสเซอร์ที่  
 เครื่องนั้นขยายได้โดยขึ้นกับการออกแบบของเครื่องแต่ละรุ่น

**3.4.3 Massively Parallel Processing Server (MPP)**

เครื่องจะเชื่อมกับอาร์เรย์ของโพรเซสเซอร์ผ่านถึงความเร็ว และแบนด์วิดท์สูง แต่ละโหนดคือ  
 เซิร์ฟเวอร์ในตัวของมันเอง (ซึ่งอาจเป็น SMP ก็ได้) ประกอบด้วยโพรเซสเซอร์ และหน่วยความ  
 จำภายในในแต่ละโหนด เพื่อ optimize สถาปัตยกรรมแบบ MPP แอปพลิเคชันจะต้องทำงานแบบ  
 ขนานด้วย อย่างไรก็ตามหากเลือกใช้สถาปัตยกรรมแบบนี้ DBMS ที่เลือกใช้จะต้องเป็นเวอร์ชันที่  
 สนับสนุนการทำงานแบบขนานด้วย



รูปที่ 3.5 สถาปัตยกรรมของเซิร์ฟเวอร์

**3.5 สถาปัตยกรรมของระบบจัดการฐานข้อมูล (DBMS)**

**3.5.1 Relational Database (RDBMS)**

ในระยะเริ่มต้น Data Warehouse ได้ประสบความสำเร็จไปแล้วกับระบบจัดการฐานข้อมูล  
 แบบ RDBMS พร้อมใช้งานร่วมกับระบบช่วยในการตัดสินใจ และแอปพลิเคชันแบบไคลเอนต์/

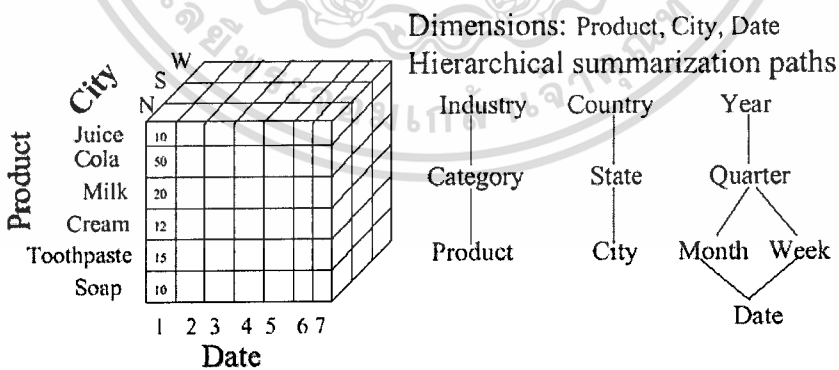
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์ โดยใช้ประโยชน์ของ RDBMS ด้านความอ่อนตัวและความสามารถในการเรียกค้นแบบทันที (ad hoc query)

RDBMS มีความอ่อนตัวที่สุดเมื่อถูกใช้กับโครงสร้างข้อมูลผ่านการ Normalized แล้ว ในฐานข้อมูลที่ normalized แล้วโครงสร้างข้อมูลจะไม่มีซ้ำซ้อน แสดงด้วย entity และความสัมพันธ์ของ entity ที่ให้รายละเอียดโดยข้อมูล (เช่น ผลิตภัณฑ์ ตลาดและทรานแซกชันการขาย) แต่คำสั่งเรียกค้นข้อมูล OLAP ทั่วไปที่ข้ามโครงสร้างหลายโครงสร้าง (อย่างคำสั่งเรียกค้น “มีอะไหล่ A กี่ชิ้นที่ขายไปเมื่อนับเฉพาะร้านที่อยู่ในภาคตะวันออก ?”) จำเป็นต้องมีโอเปอเรชัน join จำนวนมากเพื่อรวมข้อมูลเข้าด้วยกัน ขณะที่ประสิทธิภาพของ traditional RDBMS จะดีกว่าสำหรับคำสั่งเรียกค้นโดยใช้ key (เช่น “หาลูกค้าที่มีเลขที่บัญชี 1234”) มากกว่าคำสั่งเรียกค้นที่ค้นจากเนื้อหาข้อมูลเอง (เช่น “หาลูกค้าที่มีรายได้มากกว่า 50,000 เหรียญสหรัฐฯที่ซื้อรถเมื่อหกเดือนที่ผ่านมา”)

### 3.5.2 Multidimensional Database (MDDDB)

Multidimensional Database มีโครงสร้างการจัดเก็บข้อมูลในลักษณะของมิติ (Dimension) ข้อมูลหรือสิ่งที่เราสนใจหนึ่งเรื่องถือเป็นหนึ่งมิติ ถ้าสิ่งที่เราสนใจมีมากกว่าหนึ่งเรื่องก็จะสามารถดูข้อมูลได้หลายมิติ ซึ่งแต่ละมิติก็จะเก็บค่าของข้อมูลเป็นตัวเลขไว้เพื่อใช้ในการคำนวณและวิเคราะห์ โดยการเก็บข้อมูลจะเก็บเป็นโครงสร้างลูกบาศก์ (Data Cube) ดังรูปที่ 3.6 ซึ่งสามารถมองเห็นเป็นสามมิติ ทำให้เห็นภาพข้อมูลได้ชัดเจนและการวิเคราะห์ข้อมูลทำได้ง่ายด้วยคุณสมบัติต่อไปนี้



รูปที่ 3.6 Multidimensional Data

- Rotation คือสามารถดูข้อมูลได้หลายมุมมอง หมุนดูข้อมูลได้จากทุกด้านของลูกบาศก์ ทำให้สามารถจัดการกับข้อมูลที่เก็บไว้ได้หลายวิธี และเห็นความสัมพันธ์ที่ข้ามมิติใดเมนชันได้ชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Ranging คือสามารถเลือกดูข้อมูลตามขอบเขตที่เราต้องการได้ หรือเรียกดูข้อมูลเฉพาะส่วนใดส่วนหนึ่ง (Slice and dice) ที่เราสนใจมาทำการวิเคราะห์ได้ โดยไม่ต้องอาศัยข้อมูลทั้งหมด
- Hierarchy ลักษณะของข้อมูลสามารถจัดแบ่งได้เป็นลำดับชั้นทำให้สามารถจัดระดับความลึกของข้อมูลได้ การวิเคราะห์ข้อมูลจึงสามารถทำได้ทั้งแบบ Drill-down และ Roll-up โดยการ Drill-down คือการดูข้อมูลจากระดับบนแล้วค่อยขอย่อยมาดูในระดับล่าง เช่น ดูยอดขายรายปีแล้ว Drill-down มาดูยอดขายในระดับรายเดือน ส่วนการดูข้อมูลแบบ Roll-up ก็ตรงกันข้ามกัน คือเริ่มดูข้อมูลจากส่วนรายละเอียดหรือดูจากระดับล่างก่อนแล้วค่อยดูขึ้นไปยังระดับบน

เครื่องมือช่วยเรียกใช้ Data Warehouse ส่วนใหญ่เริ่มหันมาใช้ความสามารถด้าน Multidimensional ของข้อมูลในคลังข้อมูล ดังตัวอย่างเช่น การวิเคราะห์ด้านการตลาดจำเป็นต้องดูยอดขายแบ่งตามผลิตภัณฑ์ ตามตลาด ตามช่วงเวลา ตามระดับการประชาสัมพันธ์และการโฆษณา และโดยอาจลองพิจารณาจากเงื่อนไขต่างๆ ผสมกันดูเพื่อเปรียบเทียบ การจัดโครงสร้างข้อมูลใน RDBMS แบบเดิมเพื่อรองรับคำสั่งค้นหา และวิเคราะห์แบบเดิมอาจไม่เพียงพอแล้ว MDDB ทำให้การจัดการกับออบเจกต์ข้อมูลแบบ Multidimensional ทำได้โดยตรง เช่น หมุนข้อมูลเพื่อดูความแตกต่างระหว่างไคเมนชันได้โดยง่าย หรือ Drill down operations ที่ให้ระดับข้อมูลที่ลึกซึ่งยิ่งขึ้น โดยเมื่อสร้างโครงสร้างข้อมูล ต้องทำการกำหนดไคเมนชันเหล่านี้ไว้ให้ตั้งแต่เริ่มต้น เพราะการเปลี่ยนแปลงเล็กๆ น้อยๆ ที่เกิดขึ้นภายหลัง อย่างเพิ่มไคเมนชันใหม่หรือเปลี่ยน view ที่ต้องการดู อาจเป็นภาระและมีค่าใช้จ่ายสูงกว่าที่คิดมากทีเดียว

### 3.5.3 Object-Relational Database

เนื่องจากข้อจำกัดของทั้ง RDBMS และ MDDB ที่ไม่สนับสนุนชนิดข้อมูลที่ไม่ได้เป็นมาตรฐาน เช่น ภาพ เอกสาร และวิดีโอหรือออดิโอคลิป หากต้องการเก็บออบเจกต์ข้อมูลเหล่านี้ใน Data Warehouse ก็ต้องใช้ระบบจัดการฐานข้อมูลแบบ Object-relational DBMS แต่ข้อดีของแนวทางนี้ก็คือการ Encapsulation ข้อมูลในชนิดข้อมูลพิเศษ เช่น ชุดของราคาหุ้นกับเวลาในแต่ละแถวของตารางหุ้น อาจต้องการโอเปอเรเตอร์พิเศษสำหรับการค้นหาข้อมูลพื่นๆ ก่อนหน้านี้ (เช่น หาหุ้นทั้งหมดที่มีราคาตลาดต่ำลงตั้งแต่เดือนเมษายนถึงพฤษภาคม 2001)

การเลือก DBMS จะเกี่ยวข้องกับฮาร์ดแวร์ของเซิร์ฟเวอร์ที่เลือกด้วย บาง RDBMS อย่าง DB2 Parallel, Informix XPS และ Oracle อันเป็นเวอร์ชันที่สนับสนุนการทำงานแบบขนาน ซอร์ฟแวร์ที่ทำงานแบบขนานจะแตกคำสั่งค้นหา joins ข้ามโพรเซสเซอร์หลายตัว และจะ run โอเปอเรชันเหล่านี้พร้อมกันเพื่อเพิ่มประสิทธิภาพ การทำงานแบบขนานจำเป็นต้องมีฐานข้อมูลที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานอย่างมีประสิทธิภาพที่สุดบน MPP ขนาดใหญ่และ clustered SMP server ซึ่งยังไม่ใช้ทางเลือกที่มีใน MDDDB หรือ Object-relational DBMS ดังตารางที่ 3.1 (Weldon. 2540 : 128)

ตารางที่ 3.1 เปรียบเทียบระหว่าง DBMS ชนิดต่างๆ

Feature/Function	Relational	Super-Relational	Multi-dimensional	Object-Relational
Normalized structures	✓	✓		✓
Abstract data types				✓
Parallelism	✓			
Multidimensional structures		✓	✓	
Drill-down			✓	✓
Rotation			✓	✓
Data-dependent operations				✓

### 3.6 การเลือกสถาปัตยกรรมเพื่อสนับสนุนการทำงานของ Data Warehouse

การตัดสินใจเลือกสถาปัตยกรรมใดๆ สำหรับ Data warehouse ไม่ว่าจะเป็สถาปัตยกรรมของโครงสร้างข้อมูล ระบบจัดการฐานข้อมูล หรือสถาปัตยกรรมของเซิร์ฟเวอร์ มีข้อควรพิจารณาในการเลือกโดยคำนึงถึงสภาพแวดล้อมต่างๆ ดังตารางที่ 3.2

การอิมพลีเมนต์ Data Warehouse แปรเปลี่ยนไปตามสภาพแวดล้อมต่างๆ บางตัวอาจถูกออกแบบมาให้สนับสนุนการวิเคราะห์ความต้องการเฉพาะสำหรับแต่ละแผนก หรือแต่ละฟังก์ชันการทำงานขององค์กร เช่น แผนกการเงิน การตลาดหรือการขาย ขณะที่การอิมพลีเมนต์อื่นอาจเน้นไปถึงการดึงข้อมูลจากส่วนต่างๆ ขององค์กรเข้ามารวมกันเพื่อสนับสนุนกลุ่มผู้ใช้ หรือการใช้งานฟังก์ชันที่แตกต่างกันไป ยิ่งออกแบบให้ Data Warehouse มีขอบข่ายการทำงานมากขึ้น ก็จำเป็นต้องใช้พลังประมวลผลของเซิร์ฟเวอร์และ DBMS มากขึ้นตามไปด้วย นอกจากตัวแปรต่างๆ ในตารางข้างต้นแล้ว ก็ยังมีสิ่งที่ควรพิจารณาประกอบกันเพื่อให้ได้สถาปัตยกรรมที่เหมาะสมต่อการใช้งานได้แก่

- รูปแบบการใช้งานของ Data Warehouse อย่างการสอบถาม (queries) และรายงาน (report) จะเป็นตัวแปรในการเลือก view ที่ถูกจัดเตรียมไว้ล่วงหน้า มักจะให้ข้อมูลที่ผู้ใช้พึงพอใจ ขณะที่ไม่ได้ใช้พลังประมวลผลของเซิร์ฟเวอร์และ DBMS มากนัก ส่วนการวิเคราะห์ที่ซับซ้อนจะต้องการพลังการประมวลผลและความอ่อนตัวของเซิร์ฟเวอร์สูง การเรียกค้นข้อมูลจาก Data เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Warehouse ที่มีขนาดใหญ่หลายๆ เหมาะที่จะใช้งานกับ DBMS และเซิร์ฟเวอร์ที่มีความสามารถทำงานแบบขนาน

- การรับมือกับสภาพแวดล้อมที่มีการเปลี่ยนแปลงตลอดเวลา ต้องไม่ซับซ้อนและมีสถาปัตยกรรมข้อมูลที่เปลี่ยนแปลงได้ง่าย มากกว่าที่จะเป็นโครงสร้างข้อมูลที่ซับซ้อนที่ต้องการ rebuilding หลังจากการเปลี่ยนแปลงแต่ละครั้ง (เช่น โครงสร้างแบบ Multidimensional) data-refresh rate ที่ต้องการแสดงให้เห็นว่า Data Warehouse ต้องเปลี่ยนหรือจัดข้อมูลใหม่ถี่แค่ไหน จำนวนข้อมูลขนาดมหาศาลที่ต้องรีเฟรชบ่อยครั้งจะเหมาะกับสถาปัตยกรรมแบบรวมศูนย์ เพราะมีประสิทธิภาพในการ data capture และลดเวลาส่งผ่านข้อมูลลง

- ข้อมูลรายละเอียดของผู้ใช้จะต้องบอกได้ว่าใครเป็นผู้ใช้ และข้อมูลเกี่ยวกับความต้องการใช้งานข้อมูลใน Data Warehouse ของผู้ใช้แต่ละกลุ่ม จะช่วยให้สามารถวิเคราะห์รูปแบบการใช้งานหลากหลายได้อย่างถูกต้อง

- ข้อมูลที่ว่า Data Warehouse ตั้งอยู่ที่ใด จำเป็นต้องดูแลรักษาอย่างน้อยเพียงใด จะช่วยให้ตัดสินใจได้ว่าข้อมูลใน Data Warehouse จะกระจายและมีขอบข่ายเพียงใด

- ปริมาณผู้ใช้ และรูปแบบการใช้งานจะช่วยในการตัดสินใจเรื่องขนาดของเซิร์ฟเวอร์ ขนาดหน่วยความจำ และ I/O throughput ที่จะรองรับจำนวนผู้ใช้ที่เข้ามาเรียกใช้พร้อมกันในสภาวะปกติและใช้ช่วงเวลาที่มีการใช้งานสูงสุดขององค์กร

ดังนั้นสิ่งสำคัญในการเลือกส่วนผสมของสถาปัตยกรรมเซิร์ฟเวอร์และระบบจัดการฐานข้อมูลของ Data Warehouse ให้ถูกต้อง ผู้พัฒนาต้องเข้าใจความต้องการของธุรกิจ จำนวนผู้ใช้ และทักษะของทีมซัพพอร์ตด้วยว่ามีศักยภาพเพียงใด รวมถึง IS resources ที่มีอยู่ เพื่อเป็นตัวกำหนดความซับซ้อนและความลึกซึ้งของสถาปัตยกรรมต่างๆ ที่เหมาะสมมาประกอบกันเป็น Data Warehouse อีกทั้งไม่ว่าจะเป็นการเลือกขนาด ประสิทธิภาพ และชนิดของฮาร์ดแวร์และซอฟต์แวร์ให้เหมาะสมกับราคาและงานที่ต้องใช้ ก็ต้องใช้การตัดสินใจอย่างรอบคอบด้วยเช่นกัน หากเลือกผิดพลาด Data Warehouse ก็จะกลายเป็นปัญหาระดับองค์กรที่ยากต่อการใช้งาน ลื่นเปลืองค่าใช้จ่ายในการดูแล และแก้ไขจำนวนมาก จนสุดท้ายต้องล้มเหลวและเลิกใช้ไปในที่สุด

ตารางที่ 3.2 Data Warehouse Decision Matrix

For these environments...			Choose...		
Business requirements	User population	Systems support	Architecture	Server	DBMS
<b>Scope :</b> departmental <b>Uses :</b> data analysis	Small; Single location	Minimal local; Average central	Consolidated; Turnkey package	Single-processor or SMP	MDDB
<b>Scope :</b> departmental <b>Uses :</b> analysis plus informational	Large; analysts at single location; informational users dispersed	Minimal local; Average central	Tiered; detail at central; Summary at local	Clustered SMP for central; SP or SMP for local	RDBMS for central; MDDB for local
<b>Scope :</b> enterprise <b>Uses :</b> analysis plus informational	Large; geographically dispersed	Strong central	Centralized	Clustered SMP	Object- relational; Web support
<b>Scope :</b> departmental <b>Uses :</b> exploratory	Small; few sites	Strong central	Centralized	MPP	RDBMS with parallel support

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### เครื่องมือวิเคราะห์ข้อมูล

ข้อมูลใน Data Warehouse มีประโยชน์สำหรับนักวิเคราะห์ที่จะใช้ในการตัดสินใจ เป็นการตัดสินใจโดยอิงจากข้อมูลในอดีตเพื่อตอบคำถามว่าอะไรเกิดขึ้นในอดีต และคาดการณ์ว่าอะไรจะเกิดขึ้นในอนาคต การจะทำเช่นนั้นได้นักวิเคราะห์จะต้องเริ่มจากการตั้งสมมติฐาน และค่อยทำการตรวจสอบข้อมูลใน Data Warehouse ว่าสมมติฐานเป็นจริงหรือไม่ และคำถามอื่นๆ จะเกิดขึ้นมาก็ตรวจสอบไปเรื่อยๆ จนได้ข้อสรุปและตัดข้อสงสัยที่ไม่เป็นความจริงออกไปได้ และถ้านักวิเคราะห์เป็นคนช่างสังเกตก็อาจจะพบความสัมพันธ์ระหว่างข้อมูลที่ไม่คาดคิดมาก่อนก็ได้

ปัจจุบันมีเครื่องมือช่วยดึงข่าวสารทางธุรกิจความสามารถสูงหลายประเภทซึ่งทำงานอยู่เหนือ Data Warehouse โดยเครื่องมือเหล่านี้จะช่วยชี้ความสำเร็จหรือล้มเหลวของ Data Warehouse ได้เป็นอย่างดีเพราะผู้ใช้จะได้ข้อมูลตามความสามารถของเครื่องมือที่เลือกใช้นั่นเอง เครื่องมือดังกล่าวได้แก่

#### 4.1 เครื่องมือในการสอบถามและทำรายงาน (Querying and Reporting Tools)

เป็นเครื่องมือสำหรับใช้ในการเรียกค้นข้อมูลและจัดทำรายงาน เป็นการทำงานพื้นฐานที่ง่ายที่สุดสำหรับผลิตภัณฑ์เหล่านี้ ทุกตัวจะมีกราฟฟิกรูปร่างเอ็นดีสำหรับสร้างคำสั่ง SQL แทนที่ผู้ใช้ต้องเรียนรู้ภาษา SQL เองหรือเขียน โปรแกรมเองเพื่อเรียกใช้ข้อมูลในฐานข้อมูล เครื่องมือเรียกค้นจะทำหน้าที่เรียกใช้ฐานข้อมูล ดึงข้อมูลที่เกี่ยวข้องขึ้นมา คำนวณเพิ่มเติมและจัดข้อมูลเสียใหม่ถ้าจำเป็น รวมถึงแสดงข้อมูลทั้งหมดออกมาในฟอร์มเมตที่เข้าใจได้ง่ายขึ้น ทัวไปแล้วการประมวลผลทางสถิติจะจำกัดอยู่แค่การหาค่าเฉลี่ย ผลรวม ความเบี่ยงเบนมาตรฐานและฟังก์ชันวิเคราะห์ข้อมูลพื้นฐานบางตัวเท่านั้น แม้ว่าความสามารถของผลิตภัณฑ์แต่ละตัวจะแตกต่างกันไปบ้าง แต่เครื่องมือเรียกค้นและออกรายงานส่วนใหญ่ก็จะเหมาะกับคำถามประเภทที่ว่า “มีอะไรเกิดขึ้น” หรือ ยอดขายของสินค้า X, Y และ Z เมื่อเดือนที่แล้วเป็นอย่างไร เมื่อเปรียบเทียบกับเดือนก่อนหน้านั้น หรือเมื่อเทียบกับเดือนเดียวกันในปีที่ผ่านมา

เพื่อสร้างคำสั่งเรียกค้นให้เหมาะสมกับผู้ใช้ที่ไม่มีพื้นฐานเทคนิคมากขึ้นผลิตภัณฑ์อย่าง

Crystal Reports (Seagate), Impromptu (Cognos), ReportSmith (Borland), Intelligent Query (IQ) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Software), Esperant (Software AG), GQL (Andyne) จัดให้มีกราฟิกยูสเซอร์อินเทอร์เฟซที่ผู้ใช้เพียงแค่ลากและปล่อยก็ใช้งานได้แล้ว และการเรียกใช้ข้อมูลจะถูกปรับปรุงขึ้นในผลิตภัณฑ์ของผู้ขายเวอร์ชันใหม่ๆ ผู้ขายจะออกมาตรฐานไครเวอร์ เช่น ODBC และ native 32-bit drivers สำหรับแหล่งข้อมูลที่เป็นที่นิยมออกมา

โดยทั่วไป ผู้ดูแล Data Warehouse ที่ใช้ผลิตภัณฑ์เหล่านี้ ต้องเสียเวลากับการเซตอัปใหม่ เช่น การจัดการไดเรกทอรีและไลบรารีใหม่ ติดตั้งซอฟต์แวร์เชื่อมต่อ ตั้งชื่อในรูปภาษาอังกฤษ และเตรียมคำนวณ virtual data fields ไว้ล่วงหน้า เมื่อสร้างพรีอนเอ็นด์ SQL แล้วอาจจำเป็นต้องพัฒนาชุดคำสั่ง และรายงานมาตรฐานที่ต้องการ แม้ว่าบางผลิตภัณฑ์จะเสนอไลบรารีเทมเพลตที่สร้างไว้ให้แล้ว หรือฟอร์มรายงานที่ผู้ใช้สามารถปรับเปลี่ยนได้อย่างรวดเร็วก็ตาม

#### 4.2 On-line Analytical Processing (OLAP)

สำหรับผู้ใช้ที่ตัวสร้างรายงานอาจไม่สนองตอบความต้องการที่มากกว่า view ข้อมูลแบบตายตัวได้ OLAP tools ที่มี Slice and Dice Capabilities ที่สามารถแบ่งข้อมูลออกมาแสดงในตำแหน่งและรูปแบบต่างๆ ได้มากกว่า จะตอบคำถามที่มากกว่าแค่ “มีอะไรเกิดขึ้น” ด้วย OLAP tools จะช่วยให้ผู้ใช้วิเคราะห์ได้ว่าทำไมผลลัพธ์จึงออกมาอย่างนี้

เนื่องจากธรรมชาติของข้อมูลในทางธุรกิจมีลักษณะเป็นหลายมิติ อย่างเช่น ข้อมูลยอดขาย ก็มีลักษณะเป็นยอดขายต่อสินค้า ต่อเวลา และต่อสถานที่ที่กำหนดสินค้า เราเรียกว่าข้อมูลมีหลายมุมมอง เราเรียกมุมมองเหล่านี้ว่าเป็นโดเมนชั้นของข้อมูล สินค้า เวลา และสถานที่ที่กำหนดถือเป็นโดเมนชั้นของข้อมูลยอดขาย โดเมนชั้นเองก็มีลักษณะเป็นลำดับชั้น (Hierachy) คือมีการแบ่งเป็นลำดับชั้น อย่างเช่น สินค้าเป็นระดับล่างสุด ขึ้นไปเป็นขนาดของสินค้า และเป็นประเภทของสินค้า หรืออย่างเวลาก็เริ่มระดับล่างสุดเป็นวันที่ ขึ้นไปเป็นสัปดาห์ เดือนและปี หรืออย่างสถานที่ที่กำหนดสินค้า ระดับล่างสุดเป็นจุดขายสินค้า ขึ้นไปเป็นตำบล อำเภอและจังหวัด เป็นต้น การวิเคราะห์ข้อมูลในลักษณะดังกล่าวมีคำศัพท์ที่เกี่ยวข้อง 3 คำคือ (อนุชิต สุเมธวิทย์. 2541 :144)

- **Drill-down** หมายถึงการวิเคราะห์ข้อมูล โดยดูจากข้อมูลในไฮราคีระดับบนแล้วค่อยชอยย่อยมาดูในระดับล่างซึ่งละเอียดขึ้น เช่น ดูยอดขายรายปี แล้ว Drill-down มาดูยอดขายในระดัทรายเดือน เป็นต้น

- **Roll-up** มีลักษณะตรงข้ามกับ Drill-down คือเป็นการวิเคราะห์ข้อมูล โดยดูจากข้อมูลจากส่วนรายละเอียดหรือข้อมูลในระดับล่าง แล้วค่อยดูข้อมูลในระดับบน เช่น ดูยอดขายรายเดือน แล้ว Roll-up มาดูยอดขายรายปี เป็นต้น

- **Slice and Dice** หมายถึงการมองดูข้อมูลจากหลายๆ มุมและหมุนไปมา เช่น ดูยอดขายต่อเดือน แล้วอาจเปลี่ยนมาดูยอดขายต่อจังหวัดต่อปี เป็นต้น

ความจำเป็นในการวิเคราะห์ข้อมูลในลักษณะดังกล่าวทำให้เกิดเทคโนโลยีที่เรียกว่า OLAP หรือ On-line Analytical Processing ซึ่งมีความสามารถในการค้นหาและวิเคราะห์ข้อมูลจากฐานข้อมูลที่มีขนาดใหญ่ รองรับการวิเคราะห์ข้อมูลที่มีปริมาณมากๆ และตอบสนองความต้องการของผู้ใช้ได้อย่างรวดเร็ว มีสองแนวทางสำหรับ OLAP ให้เลือกใช้ ดังนี้ (Simon, 2540 : 107-108)

#### 4.2.1 MOLAP หรือ Multidimensional OLAP

มีพื้นฐานบน Multidimensional database (MDDDB) ซึ่งเก็บข้อมูลในโครงสร้างแบบลูกบาศก์หรืออาเรย์แบบหลายมิติ ทำให้สามารถจัดการกับข้อมูลที่เก็บได้หลายวิธี และเห็นความสัมพันธ์ที่ข้ามมิติใดเมมชันได้ชัดเจน แต่แนวทางนี้มีข้อจำกัดอยู่สองประการ หนึ่งคือโครงสร้างเก็บข้อมูลใหม่ต้องการฐานข้อมูลเฉพาะยี่ห้อ ที่การเรียกใช้ข้อมูลมิติใดเมมชันแนลไม่เป็นไปตามมาตรฐานระบบเปิดอย่างแท้จริง ข้อจำกัดที่สองคือโดยทั่วไปแล้วข้อมูลที่มีอยู่ในองค์กร มักเก็บอยู่ในรูปแบบของ Relational Database นั้นหมายความว่า ถ้าจะนำข้อมูลมาใช้ใน MDDDB จะต้องดึงข้อมูลฟอร์แมตและโหลดข้อมูลเหล่านี้ลงใน Hypercube ซึ่งกระบวนการนี้ซับซ้อนและกินเวลามาก แต่ผู้ขายก็พบทางออกโดยใช้ Data pump และเครื่องมืออื่นๆ ดำเนินกระบวนการนี้ให้อัตโนมัติโดยการแมปฟิลด์ข้อมูลในรีเลชันแนลไปสู่โครงสร้างมิติใดเมมชันแนล และย้ายข้อมูลไปสู่ MDDDB ขณะทำงาน ทำให้แนวทางนี้ดูจะเป็นไปได้มากขึ้น

#### 4.2.2 ROLAP หรือ Relational OLAP

มีพื้นฐานบน Relational Database (RDBMS) โดยจัดเก็บข้อมูลในโมเดลที่ออกแบบมาเฉพาะเรียกว่า Star Scheme โดยมีตารางหลัก (Fact table) อยู่ตรงกลางและมีตารางใดเมมชัน (Dimension table) ล้อมรอบ ส่วน Snowflake Scheme ก็จะมีการแยกตารางใดเมมชันออกมาเป็นหลายๆ ตาราง ลักษณะการออกแบบดังกล่าวก็เพื่อต้องการเลียนแบบลักษณะมิติใดเมมชันแนลนั่นเอง โดยการเรียกค้นข้อมูลจะทำโดยใช้คำสั่งเรียกค้น SQL มาตรฐานได้โดยตรง กลไก ROLAP จะแปลงคำสั่งเรียกค้นไปเป็น Routine SQL เลย ดังนั้นผลลัพธ์ที่ได้อาจเป็นข้อมูลจากหลายตารางอย่างสเปรคซิดมิติใดเมมชันแนล หรือรูปแบบข้อมูลที่สามารถหมุนดูรายละเอียดต่างๆ ได้ตามต้องการ

OLAP มีส่วนคล้ายกับเครื่องมือประเภท Querying and Reporting ที่ใช้กันอยู่ แต่จะทำงานได้ซับซ้อนกว่า การเซต OLAP และซอฟต์แวร์เรียกใช้ข้อมูล โมเดลข้อมูลของบริษัท และฟังก์ชันวิเคราะห์ข้อมูลต่างๆ ที่ต้องการจะต้องมีความชัดเจน โดยอาจสำรวจฟังก์ชันวิเคราะห์ข้อมูลที่ผู้บริหาร ผู้จัดการและนักวิเคราะห์ข้อมูลต้องการ โดยทั่วไปแล้ว OLAP มีลักษณะเป็น Turnkey เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

solution มากกว่าจะเป็นผลิตภัณฑ์สำเร็จรูป ที่สำคัญจะต้องสนับสนุนข้อมูลที่มีอยู่และความต้องการที่มีได้อย่างเหมาะสม ถ้าเป็นเช่นนี้แล้วระบบ OLAP จะช่วยลดภาระการดูแลผู้ใช้ทั่วไปลง ตัวอย่างผลิตภัณฑ์ประเภทนี้ ได้แก่ PowerPlay ของ Cognos, PowerPlay ของ Business Objects, BrioQuery ของ Brio Technology และ Essbase ของ Arbos Software

### 4.3 Executive Information Systems (EIS)

แนวคิดง่าย ๆ ของ Executive Information System หรือ EIS ก็คือระบบที่ให้บริการข้อมูลแก่ผู้บริหารส่วนใหญ่ที่ไม่มีเวลามากนัก ในขณะที่การวิเคราะห์ข้อมูลขนาดใหญ่ด้วย OLAP เป็นหน้าที่ของผู้เชี่ยวชาญ ผู้บริหารที่ไม่สันถัดด้านเทคนิคสามารถใช้ EIS ช่วยรวบรวมข้อมูลที่จำเป็นเข้าด้วยกัน และแสดงผลที่สามารถทำความเข้าใจได้รวดเร็ว โดยสามารถจัดกลุ่มของสภาพแวดล้อมของ EIS ได้สองกลุ่มใหญ่คือ Briefing book และ Command center

Briefing book หรือหนังสือสรุป เป็นสรุปข้อมูลเพื่อการบริหารต่างๆ ในรูปเอกสารหรือสไลด์อิเล็กทรอนิกส์ที่ทำงานแบบออนไลน์ ใช้สำหรับการตัดสินใจ หรือแสดงข้อมูลสถานะปัจจุบันของโครงการสำคัญๆ ของบริษัท ในขณะที่ Command center ทั่วไปแล้วเป็นช่องที่ไปสู่อาเรย์ของรายงานที่กว้างขวาง ข่าวสารที่รวบรวมจากนิวส์กรุปในอินเทอร์เน็ต ตลอดจนองค์ประกอบอื่นใดที่ช่วยให้องค์กรได้เข้าใจภาพรวมได้ดีขึ้น ผู้ใช้อาจกำหนดเรียกดูรายงาน Command center เป็นรายวันหรือบ่อยกว่านั้นก็ได้ ตามความถี่ในการอัปเดตข้อมูล หรือกระทั่งบางผลิตภัณฑ์จะเตือนให้ผู้ใช้ทราบเมื่อถึงกำหนดที่เซตเอาไว้

ข้อมูลที่ผู้บริหารได้จะแสดงให้เห็นภาพรวม ต่างจากโมเดล OLAP ที่มีระดับของข้อมูลเพิ่มขึ้น เหมาะกับนักวิเคราะห์ที่มีหน้าที่จัดการกับข้อมูล ผู้บริหารจะเข้าใจข้อมูลสรุปได้โดยง่าย และเมื่อผู้บริหารต้องการข้อมูลเพิ่มเติมจาก Briefing book slides หรือ Command center screens ก็สามารถทำได้ แต่หากไม่มีเวลาดูรายละเอียดก็สามารถทำได้โดยไม่พลาดข้อมูลสำคัญแต่อย่างใด นอกจากนี้สำหรับผู้บริหารที่รู้ด้านเทคนิค หรือผู้ดูแลระบบอาจดัดแปลง EIS ตามความต้องการพิเศษได้อีกด้วยการเขียนโปรแกรมหรือสคริปต์เพิ่มเติม ที่ EIS ทั่วไปจะมีมาให้ด้วย ตัวอย่างผลิตภัณฑ์ประเภทนี้ ได้แก่ Commander EIS ของ Comshare, Pilot Command center ของ Pilot Software

### 4.4 Data Mining

Data Mining เป็นกลุ่มของเครื่องมือ Open-ended analysis แทนที่จะถามคำถาม ผู้ใช้สามารถใช้เครื่องมือเหล่านี้เจาะลึกคำตอบที่ซ่อนอยู่ในข้อมูลลงไปอีก รวมทั้งแนวทาง หรือกลุ่มข้อมูลพิเศษ

ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

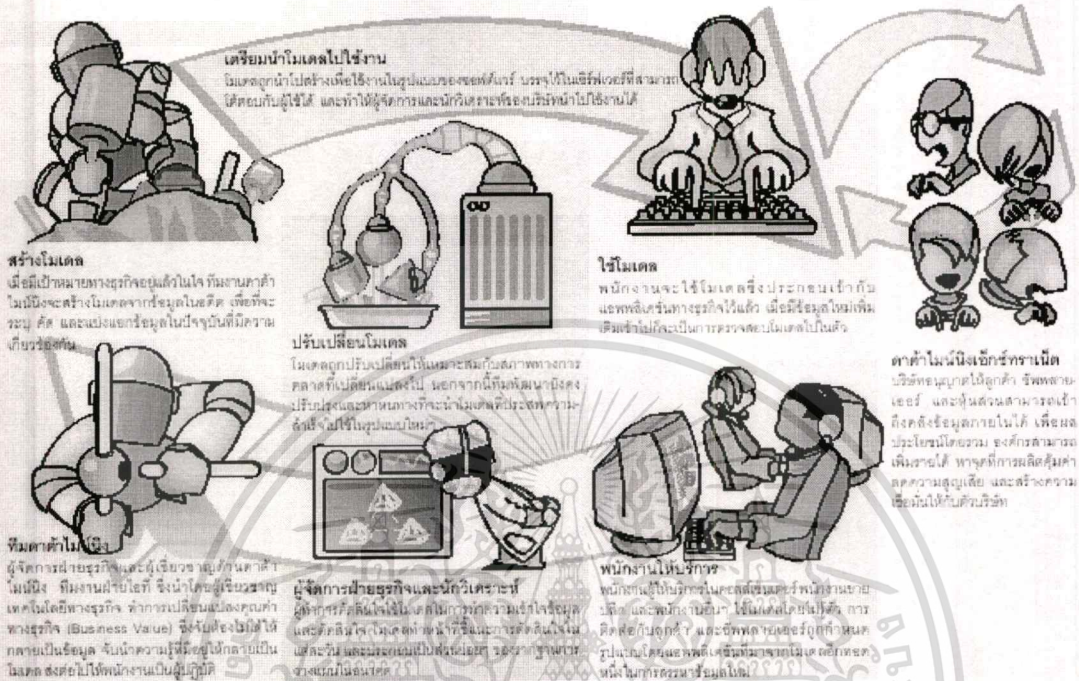
ที่สนใจ ด้วย Data Mining ทำให้สามารถดึงข่าวสารที่ซ่อนอยู่ในข้อมูล หรือคาดเดาข่าวสารจากข้อมูลที่เกี่ยวข้องใน Data Warehouse โดยไม่ต้องใช้คำถามหรือรีเควสต์เฉพาะเจาะจงลงไป เครื่องมือประเภทนี้จัดเป็น Data-driven Approach มากกว่าจะเป็น Application-driven

เทคโนโลยีฐานข้อมูลเป็นองค์ประกอบสำคัญในการส่งข้อมูลให้กับ Data Mining เพื่อใช้ในการวิเคราะห์ ส่วนเครื่องมือทางด้าน OLAP ช่วยให้การสร้างรูปแบบของ Data Mining ทำได้ง่ายยิ่งขึ้น เพราะมันจะช่วยให้เห็นมุมมองของข้อมูลที่ถูกดึง มองข้อมูลได้จากหลายๆ มุม และสามารถสลับสับเปลี่ยนข้อมูลเพื่อที่จะมองดูรูปแบบหรือแนวโน้มบางอย่างได้ชัดเจนยิ่งขึ้น เช่น อาจพบว่ายอดขายในเดือนกุมภาพันธ์สูงกว่าเดือนอื่นๆ ถึงสองเท่า ซึ่งแสดงว่าจำเป็นต้องหามาตรการบางอย่างเพื่อเพิ่มยอดขายในเดือนอื่นๆ ด้วย ซึ่งถึงแม้ว่า OLAP จะช่วยให้ผู้ใช้มองเห็นรูปแบบของข้อมูลได้ง่ายกว่าเก่า แต่ผู้บริหารก็ยังคงต้องจัดการอะไรบางอย่างกับข้อมูลที่ได้มาอยู่ดี เช่น เมื่อได้รูปแบบของข้อมูลมาแล้ว ก็จะต้องนำมาตัดสินใจต่อว่ารูปแบบใดบ้างที่มีความสำคัญ ซึ่งก็มีความเป็นไปได้สูงที่จะมองข้ามรูปแบบที่มีความสำคัญไป ไม่ว่าจะเพราะตกหล่นหรือสังเกตไม่เห็นก็ตาม และในส่วนนี้เอง Data Mining จะเป็นโพรเซสอัตโนมัติในการดึงเอารูปแบบที่ซุกซ่อนอยู่ในข้อมูลออกมาทำให้มองเห็นความเกี่ยวข้องและความสัมพันธ์กันของข้อมูล ที่ดูเผินๆ อาจจะไม่มีอะไรน่าสนใจ สิ่งเหล่านี้เป็นส่วนที่ช่วยสนับสนุนการตัดสินใจในโครงการใหญ่ๆ ซึ่งเป็นหัวใจสำคัญของการทำธุรกิจ โดยเนื้อแท้แล้วการทำงานของ Data Mining ไม่ใช่เป็นการมองที่จุดใดจุดหนึ่งของข้อมูล แต่จะเริ่มจากคำถามหรือสมมติฐานก่อน จากนั้นก็เริ่มหารูปแบบในคลังข้อมูลเพื่อมาสนับสนุนคำถามหรือสมมติฐานนั้นๆ ดังรูปที่ 4.1 (สุปรียา หล่อกิตติวิชัย. 2543 :178)

ซอฟต์แวร์ Data Mining จะใช้อัลกอริทึมที่มีความซับซ้อนหลายตัว เช่น Neural network, Rule induction, Decision tree, Predictive modeling และ Genetic algorithms ซึ่งซอฟต์แวร์ส่วนใหญ่จะวิเคราะห์ข้อมูลเป็นส่วนๆ เพื่อหากฎและรูปแบบของข้อมูล จากนั้นก็จะพิสูจน์ผลที่ได้โดยการทดสอบย้อนกลับ ซึ่งในทางวิทยาศาสตร์แล้วการทำเช่นนี้จะแสดงความสัมพันธ์ที่เดิมเป็นสิ่งที่ไม่สามารถสังเกตเห็นได้ และสกัดข้อมูลที่เป็นประโยชน์จริงๆ ออกมาจากข้อมูลดิบที่มีอยู่ และเมื่อนำมาประยุกต์ใช้กับระบบธุรกิจ สิ่งที่ได้ก็สามารถนำมาใช้ในการกำหนดนโยบาย เลือกใช้งานรูปแบบที่มีความสำคัญ รวมทั้งหลีกเลี่ยงรูปแบบที่จะก่อให้เกิดความผิดพลาดขึ้น ตัวอย่างผลิตภัณฑ์ประเภทนี้ ได้แก่ Intelligent Miner ของ IBM, Data Mine ของ Red Brick และ Visualization suite ของ Information Builder

### ระบบดาต้าไมนิงของบริษัท

คุณค่าทางธุรกิจที่แท้จริงของข้อมูล จะเกิดขึ้นเมื่อมีการนำเทคโนโลยีดาต้าไมนิงมาใช้ในทางปฏิบัติอย่างจริงจัง เพื่อช่วยให้องค์กรเป้าหมายทางธุรกิจ และโมเดลของข้อมูลที่มีธุรกิจเป็นแกนกลางนั้น ยังช่วยในการทำงานของผู้จัดการ พนักงานของบริษัท ลูกค้า ซัพพลายเออร์ และหุ้นส่วนทางธุรกิจรายอื่นๆ ด้วย



รูปที่ 4.1 ระบบ Data Mining กับการนำไปใช้งาน

### 4.5 แนวทางการเลือกเครื่องมือวิเคราะห์ข้อมูล

อาจกล่าวได้ว่าหากขาดเครื่องมือช่วยให้ผู้ใช้เรียกข้อมูลและวิเคราะห์ข้อมูลได้อย่างเหมาะสมก็ไม่มีประโยชน์อะไรที่จะอิมพลีเมนต์ Data Warehouse ขึ้นมาให้เสียเงินและเสียเวลาเปล่า สิ่งสำคัญในการเลือกเครื่องมือวิเคราะห์ข้อมูล ก็คือจงเลือกเครื่องมือที่เหมาะสมกับความต้องการของผู้ใช้ ความเชี่ยวชาญด้านเทคนิคของบุคลากรที่มีและแหล่งข้อมูลที่มีอยู่ ดังตารางที่ 4.1 เป็นการเปรียบเทียบเครื่องมือวิเคราะห์ข้อมูลประเภทต่างๆ กับความต้องการของผู้ใช้ หากเลือกผู้ขาย Data Warehouse ที่เสนอเครื่องมือรวม หรือ Integrated tools ก็จะช่วยประหยัดเวลาในการพัฒนาลงไปได้มาก หากเลือกชุดเครื่องมือคอมพิวเตอร์ที่เปิดตัวหรือเลือกชุดเครื่องมือที่สนับสนุนแหล่งข้อมูลที่มีโดยตรง

หลังจากเลือกเครื่องมือที่คอมพิวเตอร์ที่เปิดตัวกับแหล่งข้อมูลแล้ว ก็มาพิจารณาว่ามีการวิเคราะห์มากนักน้อยเพียงใดที่ต้องการอย่างแท้จริง ถ้าข้อมูลที่ต้องการเน้นที่จำนวนเป็นหลักแล้ว เครื่องมือประเภท Querying and Reporting พื้นฐานก็อาจเพียงพอแล้ว แต่ถ้าต้องการวิเคราะห์ข้อมูลอย่างซับซ้อน ที่สามารถอธิบายสาเหตุ ผลลัพธ์ที่จะเกิดขึ้น รวมทั้งแนวทางที่น่าจะเป็นแล้วก็คงต้องเลือกใช้เครื่องมือประเภท OLAP tools ซึ่งผู้ใช้อาจมองหา Relational OLAP solution ด้วยเหตุที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมแบบเปิดมากกว่า หากต้องการวิเคราะห์ข้อมูลที่ซับซ้อนและลึกซึ้งไปกว่านั้นก็อาจเลือกใช้เครื่องมือประเภท Data Mining ซึ่งผู้ใช้จำเป็นต้องมีความเชี่ยวชาญทางด้าน data analysis techniques ที่จำเป็นต่อ advanced forecasting, classification และ pattern technology

ตารางที่ 4.1 เปรียบเทียบการเลือกเครื่องมือวิเคราะห์ข้อมูลประเภทต่างๆ

ชนิดเครื่องมือ	คำถามพื้นฐาน	ตัวอย่างผลลัพธ์ที่ได้	ผู้ใช้ต้องการ
<b>Querying and Reporting</b>	มีอะไรเกิดขึ้น ?	รายงานการขายประจำเดือน, ประวัติสินค้าคงคลัง	ข้อมูลย้อนหลัง ; จำกัดเทคนิคการใช้งาน
<b>On-line Analytical Processing (OLAP)</b>	มีอะไรเกิดขึ้น ? แล้วทำไม ?	ยอดขายประจำเดือนพร้อมกับราคาเปลี่ยนแปลงของคู่แข่ง	ก้าวพ้นข้อจำกัดการดูข้อมูลแบบ static ไปสู่เทคนิค slice and dicing
<b>Excutive Information Systems (EIS)</b>	ฉันต้องรู้อะไรบ้าง ?	Briefing books; Command center	ข้อมูลสรุป และข่าวสารในระดับสูง; ไม่ใช้ความรู้ทางเทคนิค
<b>Data Mining</b>	มีอะไรน่าสนใจ ? อะไรกำลังจะเกิดขึ้นบ้าง ?	Predictive models	ดึงความสัมพันธ์และแนวทางที่ข้อมูลยังไม่แสดงชัดเจน ; ใช้เทคนิคที่ลึกซึ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

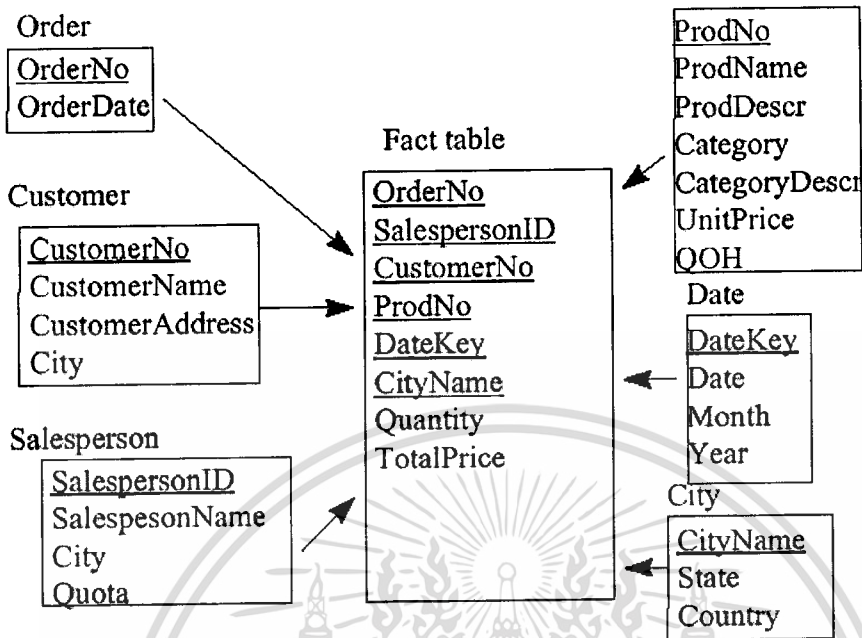
### การออกแบบฐานข้อมูลสำหรับคลังข้อมูล

การออกแบบฐานข้อมูลใน Data Warehouse แตกต่างจากการออกแบบฐานข้อมูลแบบอื่นๆ คือ การคืนORMอลไลซ์ (Denormalize) ที่ต้องหลีกเลี่ยงในงานระบบอื่นๆ กลับเป็นสิ่งที่จำเป็นใน Data Warehouse ที่ต้องการเรื่องความเร็วในการดึงข้อมูล แต่ไม่สนใจเรื่องความซ้ำซ้อน เพราะไม่ต้องการแก้ไขข้อมูลอยู่แล้ว เนื่องจากข้อมูลมีลักษณะคงที่ (Static) ไม่มีการเปลี่ยนแปลง

การออกแบบฐานข้อมูลสำหรับ Data Warehouse โดยส่วนใหญ่แล้วจะออกแบบโดยใช้ Data model แบบ Star Schema และ Snowflake Schema ซึ่งแสดงให้เห็นถึงลักษณะมิติใดเมนชันเนลของข้อมูลได้ด้วยโครงสร้างแบบง่ายๆ ไม่ซับซ้อนประกอบด้วยตารางเพียงไม่กี่ตาราง เพื่อการ join และการเรียกค้นข้อมูลที่รวดเร็ว เนื่องจากข้อมูลใน Data Warehouse จะเก็บอยู่ในรูปแบบ Denormalized form สามารถเก็บข้อมูลที่ซ้ำซ้อนกันได้ภายในตารางเดียวกัน การเก็บข้อมูลจึงไม่จำเป็นต้องแตกออกเป็นหลายๆ ตาราง ซึ่งตรงกันข้ามกับการออกแบบฐานข้อมูลโดยทั่วไป ที่ข้อมูลต้องผ่านการ Normalized เพื่อกำจัดความซ้ำซ้อน ทำให้ต้องแตกเป็นหลายๆ ตารางเพื่อเก็บข้อมูล

#### 5.1 การออกแบบฐานข้อมูลด้วย Star Schema

Star Schema เป็นเทคนิคการออกแบบที่นิยมใช้มากที่สุดในการออกแบบฐานข้อมูลของ Data Warehouse โครงสร้างการออกแบบจะประกอบด้วยตารางเพียง 2 ประเภทเท่านั้น คือ ตารางหลัก (Fact Table) และตารางใดเมนชัน (Dimension Table) โดยมี Fact table อยู่ตรงกลาง และเชื่อมกับ Dimension tables หลายๆ ตารางด้วย Dimensional key ซึ่ง Dimension table แต่ละตารางอยู่ในรูปแบบ Denormalized form



รูปที่ 5.1 Star Schema

- ตารางหลัก (Fact table)

ตารางหลักเป็นตารางในคลังข้อมูลเชิงสัมพันธ์ที่เก็บค่ารายละเอียดต่างๆ สำหรับเป็นค่าที่ใช้วัด (Measure) หรือเก็บข้อมูลเกี่ยวกับข้อเท็จจริงที่เกิดขึ้น ตารางหลักมีลักษณะโดยทั่วไป ดังนี้

- เป็นตารางขนาดใหญ่ประกอบด้วยข้อมูลเป็นจำนวนมาก ตารางประกอบด้วยข้อมูลจำนวนหลายๆแถว อาจเป็นไปได้ว่ามีถึงพันๆ ล้านเรคอร์ดก็ได้ เนื่องจากจะต้องเก็บข้อมูลในทุกๆ กิจกรรมทางธุรกิจที่เกิดขึ้น

- ข้อมูลมักเก็บอยู่ในรูปแบบตัวเลข (Numeric data)
- ข้อมูลที่เก็บจะอยู่ในลักษณะคงที่ (Static data) ไม่มีการเปลี่ยนแปลงแก้ไขใดๆ จะมีก็แต่เพียงการเพิ่มข้อมูลใหม่เข้าไปในตารางเท่านั้น

- ตารางประกอบไปด้วยหลายๆ Foreign key เพื่อใช้เชื่อมโยงไปยังตารางใดเมนชั้น

- ตารางใดเมนชั้น (Dimension table)

ตารางใดเมนชั้นเป็นตารางที่อธิบายถึงข้อมูลซึ่งถูกอ้างอิงมาจากตารางหลัก เช่น ตารางรายละเอียดสินค้า ตารางลูกค้า ตารางเวลา เป็นต้น มีลักษณะโดยทั่วไป ดังนี้

- เป็นตารางขนาดเล็ก จำนวนข้อมูลที่เก็บมีน้อยกว่าในตารางหลักมากที่สุดอาจมีเพียงร้อยหรือพันเรคอร์ดเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

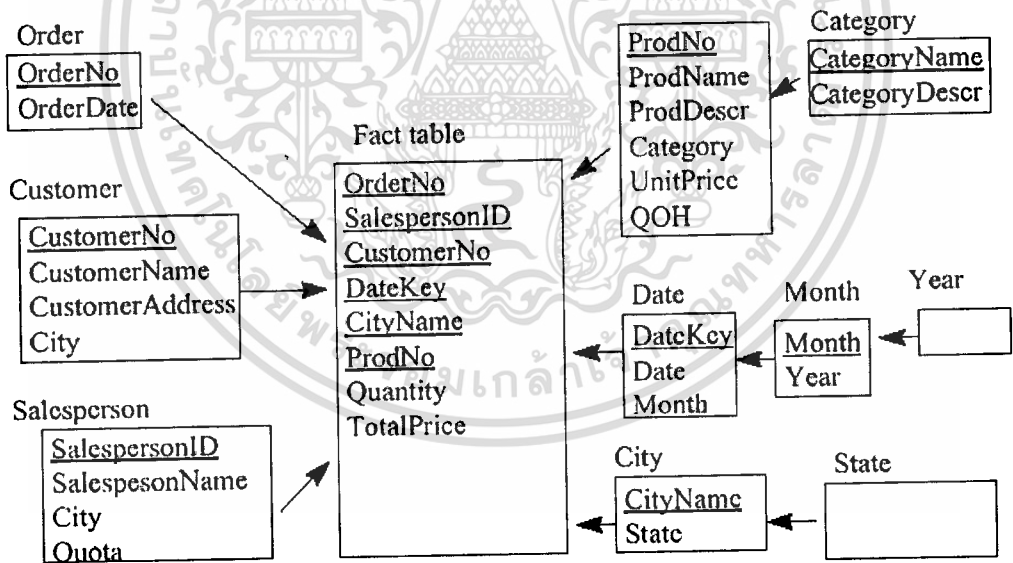
- ข้อมูลเก็บในรูปแบบตัวอักษร (Character data)
- ข้อมูลที่เก็บสามารถมีการเปลี่ยนแปลงแก้ไขได้ (Updated data) เช่น ข้อมูลเกี่ยวกับลูกค้า อาจมีการเปลี่ยนแปลงที่อยู่ หรือเบอร์โทรศัพท์ได้

- ในแต่ละตารางจะมีได้เพียง 1 Primary key หรือ Dimensional key
- สามารถจัดการข้อมูลให้อยู่ในรูปแบบที่เป็นลำดับชั้นได้ (Hierarchies)

ข้อสังเกต คือ ความสัมพันธ์ระหว่าง Foreign key ในตารางหลักกับ Primary key ใน ตารางใดเมินชั้นมีความสัมพันธ์กันแบบ many-to-one และ Primary key ของตารางหลัก จะประกอบด้วย Foreign key ต่างๆ จากทุกตารางใดเมินชั้น

## 5.2 การออกแบบฐานข้อมูลด้วย Snowflake Schema

การออกแบบจะดัดแปลงมาจาก Star Schema โดยจะนำตารางใดเมินชั้นมาทำการ Normalized ทำให้ได้ตารางแยกย่อยออกไปอีกหลายๆ ตารางเพื่อลดการเก็บข้อมูลที่ซ้ำซ้อน



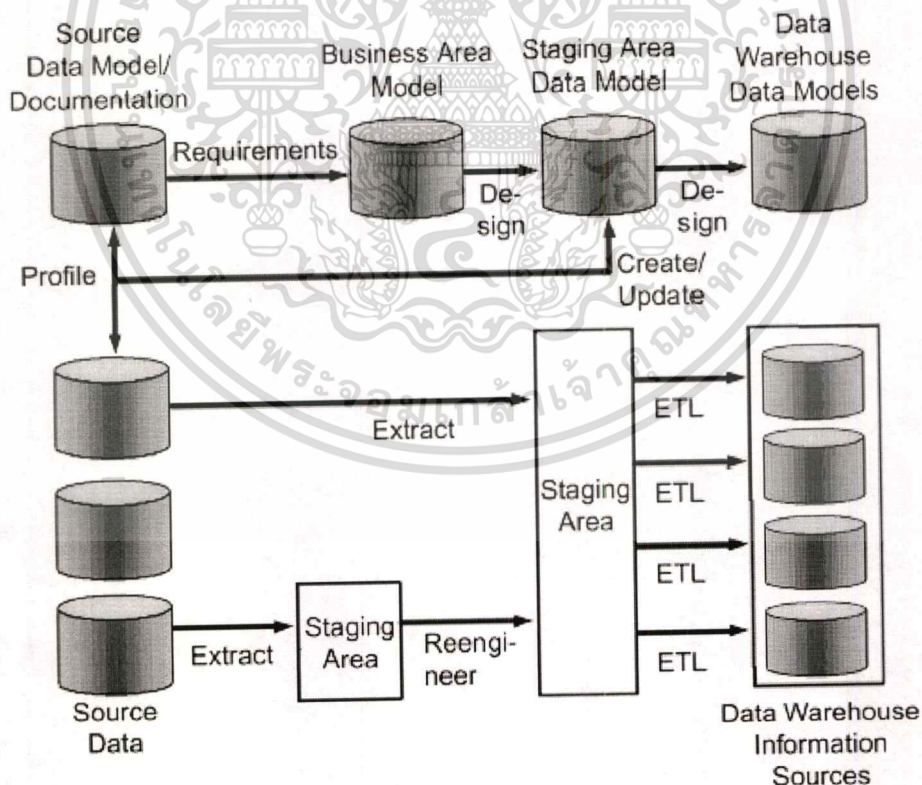
รูปที่ 5.2 Snowflake Schema

## บทที่ 6

### แนวทางการพัฒนาค้างข้อมูล

#### 6.1 ขั้นตอนการออกแบบและพัฒนาค้างข้อมูล

เนื่องจากการควบคุมคุณภาพข้อมูลเป็นสิ่งที่จะต้องให้ความสำคัญในทุกขั้นตอนของการพัฒนา Data Warehouse ดังนั้น Colin White จึงเสนอวิธีนำเครื่องมือในการทำ Data Profiling และ Data Structuring เข้ามาช่วยในการวิเคราะห์ข้อมูลก่อนที่จะรวบรวมและโยกย้ายข้อมูลเข้าสู่ Data Warehouse รวมทั้งได้อธิบายการดูแลและควบคุมคุณภาพข้อมูล ควบคู่ไปกับขั้นตอนการออกแบบและพัฒนา Data Warehouse แต่ละขั้น ดังรูปที่ 6.1 (อรุณี ชีวะเกรียงไกร. 2543ก : 115-118)



รูปที่ 6.1 ขั้นตอนการพัฒนาค้างข้อมูล

## 1. ระบุความต้องการของผู้ใช้ระบบ และออกแบบแหล่งเก็บข้อมูลสำหรับคลังข้อมูล

- การสร้างโมเดลทางธุรกิจ สำหรับความต้องการของผู้ใช้ระบบในแต่ละประเภทธุรกิจที่ Data Warehouse นั้นๆ รองรับอยู่ เป็นงานแรกที่ต้องทำในกระบวนการออกแบบและพัฒนา Data Warehouse ซึ่งความต้องการของผู้ใช้ระบบนี้ มักจะได้อาจจากการสัมภาษณ์ผู้ใช้นั้นเอง หลังจากนั้นเราก็ต้องระบุให้ได้ว่าข้อมูลทางธุรกิจ (Business Information) และกระบวนการประมวลผลเชิงวิเคราะห์ (Analytic Processing) ที่ Data Warehouse จำเป็นจะต้องรองรับได้นั้นได้แก่อะไรบ้าง สาเหตุที่เราต้องสร้างโมเดลทางธุรกิจขึ้นมา ก็เนื่องจากข้อมูลที่เกี่ยวข้องกับโมเดลทางธุรกิจนี้จะช่วยให้เราออกแบบแหล่งเก็บข้อมูลของ Data Warehouse ที่เหมาะสมกับโมเดลดังกล่าวได้นั่นเอง

- การออกแบบ Staging Area Model เป็นกุญแจสำคัญที่จะช่วยให้เราได้ข้อมูลที่มีคุณภาพในการรวบรวมข้อมูล (Integration) ในสภาพแวดล้อมของ Data Warehouse ซึ่งการดึงข้อมูลที่มาจกแหล่งข้อมูลผ่าน Staging Area นี้จะทำให้ข้อมูลที่ต้องการนำไปเก็บในแหล่งเก็บข้อมูลของ Data Warehouse เป็นข้อมูลชุดที่ผ่านการคัดสรรให้ถูกต้องตรงกัน (Consistent) กันทั้งระบบ เนื่องจากข้อมูลที่ถูกรวบรวมมาจากแหล่งข้อมูลในขั้นนี้ จะถูกตรวจสอบคุณภาพและกำจัดข้อมูลที่มีคุณภาพออกไป ก่อนที่จะนำไปเก็บในแหล่งเก็บข้อมูลต่างๆ ที่ประกอบกันขึ้นมาเป็นตัว Data Warehousing System ไม่ว่าจะเป็น Data Warehouse หรือ Data Mart ก็ตาม

## 2. เตรียมแม่แบบข้อมูลจากแหล่งข้อมูลเข้าสู่ Staging Area

เมื่อใดก็ตามที่เราเข้าใจภาพรวมของข้อมูลจากแหล่งข้อมูลเรียบร้อยแล้ว เราควรจะเริ่มจัดทำเอกสารเกี่ยวกับการแม่แบบข้อมูลที่จะดึงข้อมูลจากแหล่งข้อมูลเข้าสู่ Staging Area ซึ่งขั้นตอนนี้อาจใช้เครื่องมือทำ Data Profiling เข้ามาช่วยจัดทำให้โดยอัตโนมัติได้ โดยอาศัย Metadata ที่ได้จาก Staging Area Data Model ที่ผู้ใช้นั้นเอง ยิ่งไปกว่านั้นปัจจุบันผู้ขายแอปพลิเคชันเชิงวิเคราะห์และผู้ขายเครื่องมือ ETL มักจะเตรียมเทมเพลตสำหรับการดึงและแปลงข้อมูล ที่ประกอบด้วยการแม่แบบในรูปแบบต่างๆ ไว้ให้อยู่แล้วเพื่อใช้ในการดึงข้อมูลและโหลดข้อมูลเข้าสู่ Staging Area

## 3. ดึงข้อมูลจากแหล่งที่ต้องการและโหลดข้อมูลเข้าสู่ Staging Area

จุดประสงค์ของขั้นตอนนี้คือการดึงข้อมูลจากเพิ่มข้อมูลแต่ละเพิ่มจากแหล่งข้อมูลต่างๆ แล้วนำไปจัดรูปแบบใหม่ให้อยู่ในฟอร์มที่ตรงกับองค์ประกอบทางธุรกิจ ในแต่ละประเภทธุรกิจของคลังข้อมูลนั้นๆ สิ่งที่จะต้องให้ความสำคัญในขั้นนี้ก็คือการกำจัดข้อมูลที่ไม่มีความหมายหรือไม่มีคุณภาพเพียงพอ และการแปลงข้อมูลนั้นๆ หรือการรวมข้อมูลจากหลายๆ เพิ่มเข้าด้วยกันนั่นเอง

ในขั้นตอนนี้ก็มีเครื่องมือ ETL หลายชิ้นที่สามารถใช้ดึงและโหลดข้อมูลเข้าสู่ Staging Area แต่ทั้งนี้ทั้งนั้น วิธีการที่ใช้ในเครื่องมือ ETL แต่ละชนิดนั้นยังเป็นวิธีการที่ไม่เป็นมาตรฐานนัก กล่าวคือ ยังเป็นภาษาที่ผู้ขายแต่ละรายกำหนดขึ้นเองทั้งสิ้น (Proprietary Definition Language) ซึ่งเอกสารที่ใช้สำหรับเตรียมการแมปซึ่งถูกสร้างขึ้นในขั้นตอนนี้ก่อนหน้า จะช่วยระบุมารแมปข้อมูล ในกระบวนการดึงข้อมูล จัดรูปแบบข้อมูลใหม่และโหลดข้อมูลเข้าสู่ Staging Area

#### 4. คัดเลือก แปลงข้อมูลที่ดึงออกมาและรวม (merge) ข้อมูลนั้นเข้าสู่ Staging Area

เมื่อเราต้องการคัดเลือกข้อมูลใน Staging Area โดยอาศัยคำอธิบายข้อมูลที่ได้จากการทำ Data Profiling ข้างต้น เราจะต้องระบุมความผิดพลาดที่อาจเกิดในกฎทางธุรกิจที่มีอยู่ เกี่ยวกับความผิดพลาดเหล่านั้น และจัดการให้ข้อมูลถูกต้องตรงกันทั้งระบบให้ได้เสียก่อน ซึ่งกระบวนการเหล่านี้ล้วนเป็นงานที่ซับซ้อน โดยเฉพาะการกำจัดข้อมูลที่ไม่ต้องการมักจะมีลักษณะเฉพาะและขึ้นอยู่กับแอปพลิเคชันแต่ละชนิดทั้งสิ้น เครื่องมือที่จะช่วยเราได้ในช่วงตอนนี้ก็คือ เครื่องมือช่วยทำ Data Engineering ซึ่งมักจะรวมอยู่ในสินค้าประเภท ETL อยู่แล้ว ขั้นตอนการกำจัดข้อมูลที่ไม่มีความหมายออกไปนี้ อาจเกิดขึ้นระหว่างกระบวนการดึงข้อมูลจากแหล่งข้อมูลก็ได้ หลังจากข้อมูลถูกคัดสรรเป็นอย่างดีแล้ว เครื่องมือ ETL จึงจะสามารถแปลงและรวมข้อมูลใน Staging Area ให้อยู่ในรูปแบบที่ตรงตาม Data Model ของ Staging Area ได้

ในส่วนของการแปลงข้อมูล กระบวนการที่เกี่ยวข้องได้แก่ การถอดรหัสฟิลด์ที่เข้ารหัสไว้ การเพิ่มฟิลด์ที่เกี่ยวข้องกับเวลา (Timestamp Field) และจัดการเกี่ยวกับการตรวจสอบความถูกต้อง (Validation) ของกฎทางธุรกิจ รวมทั้งการกำจัดข้อมูลด้อยคุณภาพที่ยังเหลืออยู่ เนื่องจากเครื่องมือ ETL ไม่สามารถจัดการได้ ซึ่งในโครงการที่ไม่ซับซ้อนมากนักกระบวนการแปลงและรวมข้อมูล อาจทำในขณะที่ดึงข้อมูลได้เลย

#### 5. ดึงและแปลงข้อมูลจาก Staging Area และโหลดข้อมูลเข้าสู่แหล่งเก็บข้อมูลของ Data Warehouse

งานสุดท้ายของขั้นตอนการออกแบบ Data Warehouse คือ การแมปข้อมูล (Data Mapping) จาก Staging Area ไปที่ Data Warehouse หรือเป็นการดึงและแปลงข้อมูลที่ต้องการทั้งหมดจาก Staging Area และโหลดเข้าสู่ Data Warehouse ซึ่งการแปลงข้อมูลในขั้นตอนนี้จะเป็นการสร้างหรือรวบรวมข้อมูลสรุป มากกว่าที่จะเป็นการกำจัดข้อมูลที่ไม่มีความหมาย โดยสามารถใช้เครื่องมือ ETL ในการดึง แปลง และโหลดข้อมูลได้เหมือนกับขั้นตอนอื่นๆ

ในการแมปข้อมูลนี้ เราจะต้องระบุแหล่งข้อมูลทั้งหมดที่น่าจะมาใช้ในระบบ Data Warehouse และจัดทำคำอธิบายข้อมูล (Profiling) นั้นๆ เพื่อใช้ในการพิจารณาว่าข้อมูลที่ต้องการเอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บใน Data Warehouse สามารถดึงได้จากแหล่งข้อมูลที่ใช้ในแอปพลิเคชันทางธุรกรรม (Transactional Application) ที่มีอยู่แล้วหรือไม่ ถ้าข้อมูลที่ต้องการไม่สามารถหาได้จากแหล่งข้อมูลที่มีอยู่ เราก็จะต้องหาวิธีที่จะทำให้ได้ข้อมูลนั้นมา หรือบางครั้งอาจจะต้องปรับความต้องการของโครงการ และความคาดหวังของผู้ใช้ระบบในกรณีที่เป็น นอกจากนี้สิ่งที่ต้องทำอีกอย่างก็คือ จะต้องพิสูจน์ให้ได้ว่าแหล่งข้อมูลที่มีอยู่มีศักยภาพเพียงพอเพื่อให้โครงการอยู่ในงบประมาณและเวลาที่กำหนด และในกระบวนการแปลงข้อมูล (Conversion Process) ยังจะต้องระบุและแก้ไขความผิดพลาด (Error) ของโครงสร้างข้อมูลและเนื้อหาข้อมูลให้เร็วที่สุดเท่าที่จะทำได้ มิฉะนั้นปัญหาคุณภาพข้อมูลอาจจะมีผลกระทบในทางลบกับกระบวนการที่ต้องทำต่อเนื่องกันมา อย่างไรก็ตาม ปัญหาคุณภาพข้อมูลใน Data Warehouse บางครั้งก็ไม่สามารถตรวจพบได้ และอาจทำคุณค่าของ Data Warehouse ลดลง ถูกมองเป็นเพียงแค่ว่าแพลตฟอร์มที่สนับสนุนการตัดสินใจเท่านั้น

หลังจากที่เราดำเนินกระบวนการทั้งหมดที่กล่าวไปข้างต้น ก็จะมาถึงกระบวนการที่เราจะต้องเพิ่มเข้าไปเพื่อให้มั่นใจได้ว่าข้อมูลที่ใช้ใน Data Warehouse มีคุณภาพเพียงพอ ซึ่งสิ่งแรกที่ต้องเพิ่มเข้าไปก็คือ การจัดทำ Data Profiling หรือการทำคำอธิบายข้อมูลที่มาจากแหล่งข้อมูลต่างๆ โดยบรรยายทั้งโครงสร้างและความบูรณภาพ (Integrity) จากแหล่งข้อมูลที่มีศักยภาพเหล่านั้น จากนั้นก็เปรียบเทียบผลลัพธ์ที่ได้จากขั้นตอนนี้กับเอกสารประกอบแหล่งข้อมูลหรือ Metadata ที่มีอยู่แล้ว การทำ Data Profiling นี้รวมถึงการวิเคราะห์ข้อมูลจากแหล่งข้อมูล การทำจัดรายการหน่วยข้อมูล (Data Element) ที่มีอยู่ ตรวจสอบรูปแบบของหน่วยข้อมูลเหล่านั้น ซึ่งผลลัพธ์ที่ได้จากการทำ Data Profiling ก็คือ Metadata ที่ทำหน้าที่อธิบายเพิ่มข้อมูลที่ได้จากแหล่งข้อมูลที่มีความทันสมัยและแม่นยำนั่นเอง Metadata จะช่วยให้ทีมพัฒนาระบุเพิ่มข้อมูลที่ตรงกับความต้องการของผู้ใช้และสามารถสร้างโมเดลข้อมูลที่ต้องการแม่นยำมากที่สุดนั่นเอง

## 6.2 บทบาทของเครื่องมือช่วยในกระบวนการ ETL

วัตถุประสงค์หลักในการสร้างเครื่องมือ ETL ขึ้นมาก็เพื่อกำจัด หรือลดความต้องการในการเขียนโปรแกรมที่ทำหน้าที่ดึงและแปลงข้อมูลขึ้นเองในองค์กร ลักษณะการทำงานของเครื่องมือ ETL นี้ จะทำงานตามกฎที่นักพัฒนาที่กำหนดให้ เครื่องมือ ETL นี้มีทั้งแบบที่สร้างโปรแกรม (เขียนด้วยภาษาต่างๆ กันเช่น ภาษา C หรือ COBOL เป็นต้น) สำหรับดึงและแปลงข้อมูลพร้อมกันเป็นชุด (Batch) ตามกฎที่กำหนดไว้แล้วให้โดยอัตโนมัติ หรือแบบดึงและแปลงข้อมูลโดยแปลความหมายของกฎต่างๆ ในแบบ runtime

ประโยชน์หลักที่ได้รับจากเครื่องมือ ETL ชนิดที่สร้างโค้ดหรือโปรแกรมขึ้นมาให้เราก็คือ ตัวโปรแกรมมักจะรองรับการแปลงข้อมูลที่ซับซ้อนได้มากกว่าเครื่องมือ ETL ที่ทำงานตามความ

เอกส...  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายของกฎที่เปลือออกมาในขณะ runtime ยิ่งไปกว่านั้น เรายังสามารถนำโปรแกรมที่ได้มาปรับเปลี่ยนให้ตรงกับความต้องการของเรามากที่สุด แทนที่จะเริ่มเขียนโปรแกรมที่เฉพาะเจาะจงใหม่ทั้งหมด อย่างไรก็ตาม เครื่องมือประเภทที่สร้างโค้ดให้เราโดยอัตโนมัติ (Code-generated tool) มักจะมีข้อเสียอยู่ว่า โปรแกรมที่ได้มักจะเป็นโปรแกรมชุดขนาดใหญ่และยากต่อการบำรุงรักษา โดยเฉพาะถ้าระบบของเรายังจะต้องอัปเดตกฎในการดึงและแปลงข้อมูล และต้องการสร้างโค้ดใหม่อีกครั้งบ่อยๆ วิธีนี้ก็จะไม่เหมาะสมนัก ตัวอย่าง เครื่องมือที่สร้างโปรแกรมให้เราโดย

ทิศทางของเครื่องมือ ETL ในปัจจุบันคือ ผู้ขายได้พยายามแก้ปัญหาในอดีต โดยเสนอตัวเลือกหรือแนวทางใหม่ เพิ่มเข้าไปในเครื่องมือ ETL ชนิดที่แปลความหมายของกฎต่างๆ เพื่อแทนการสร้างโปรแกรมอัตโนมัติ ตัวเลือกดังกล่าวคือ การสร้างภาษาขึ้นมาใหม่เพื่อใช้อธิบายการทำงานในกระบวนการ ETL ซึ่งมีหลักการการทำงานคือ แทนที่จะสร้างโค้ดขึ้นมาใหม่ เครื่องมือ ETL จะแปลความหมายของคำจำกัดความที่อธิบายโดยใช้ภาษาที่กำหนดขึ้นนี้ในระหว่างรันของ ETL เอง นอกจากนี้ยังอาจมีการเพิ่มส่วนที่เป็นเทมเพลตสำหรับดึงข้อมูลจากโปรแกรมสำเร็จรูปที่ใช้ในการประมวลผลทางธุรกรรม (Transaction Processing Package) ที่สามารถปรับปรุงเปลี่ยนแปลงให้ตรงกับความต้องการเฉพาะอย่างได้เข้าไปด้วย ประโยชน์ที่ได้จากการใช้เครื่องมือ ETL ลักษณะนี้คือสามารถทำงานได้เร็วกว่า ถูกกว่า และติดตั้งง่ายกว่า เมื่อเทียบกับชนิดที่สร้างโค้ดให้อัตโนมัติ ตัวอย่างเครื่องมือ ETL ประเภทนี้ได้แก่ Informix/Ardent DataStage, IBM Visual Warehouse (ในปัจจุบันเรียก DB2 Warehouse Manager), PowerCenter ของ Informatica Corp. และ Oracle Data Mart Suites

อย่างไรก็ตาม หลายองค์กรพบว่า ไม่ว่าจะเลือกเครื่องมือประเภทใด การเลือกใช้เฉพาะ ETL เพียงอย่างเดียวก็ไม่สามารถรับประกันได้ว่าจะไม่เกิดปัญหาเกี่ยวกับคุณภาพข้อมูลในแหล่งข้อมูลที่มีอยู่ นอกจากนี้การแก้ไขข้อมูลโดยใช้เครื่องมือ ETL ยังใช้เวลาและงบประมาณค่อนข้างมาก โดยเฉพาะอย่างยิ่งถ้าแหล่งข้อมูลที่มีอยู่มีคุณภาพต่ำมาก จะยิ่งทำให้โอกาสที่จะเกิดความผิดพลาดในการแปลงข้อมูลสูงตามไปด้วย

ดังนั้นเราควรใช้เครื่องมือในการทำ Data Profiling และ Data Reengineering ช่วยในการวิเคราะห์และระบุปัญหา และกำจัดข้อมูลที่ไม่มีความหมายออกไปก่อนที่จะใช้เครื่องมือ ETL เราเรียกกระบวนการพัฒนาล้างข้อมูลแบบนี้ว่า Analysis-led Data Warehouse Development Approach ซึ่งเสนอขึ้นมาแทนที่การพัฒนาโดยใช้เครื่องมือ ETL เป็นหลักเพียงอย่างเดียว หรือที่เรียกว่า ETL-driven Data Warehouse Development Approach

### 6.3 เครื่องมือสำหรับทำ Data Profiling

เครื่องมือทำ Data Profiling นี้ถูกสร้างขึ้นมา เพื่อสนับสนุนกระบวนการพัฒนา Data Warehouse แบบที่เสนอให้วิเคราะห์ข้อมูลก่อนที่จะโยกย้ายข้อมูลไปยังระบบใหม่ ทั้งนี้เพื่อช่วยในการทำความเข้าใจข้อมูลที่ได้จากแหล่งข้อมูล และตรวจสอบคุณภาพของข้อมูลก่อนที่จะนำมาใช้งาน ถ้าไม่พบปัญหาเราก็สามารถใช้เครื่องมือสำหรับทำ Data Engineering เพื่อประมวลผลขั้นต้น และคัดเลือกข้อมูลบางส่วนก่อนที่จะนำไปประมวลผลใน ETL ได้เลย

การทำ Data Profiling เป็นการวิเคราะห์ข้อมูลจากแหล่งข้อมูลเพื่อที่จะแจกแจงกฎทางธุรกิจที่เกี่ยวข้องกับหน่วยข้อมูล และความบูรณภาพของข้อมูล ซึ่งกฎทางธุรกิจเหล่านี้สามารถที่จะนำมาใช้ในการอัปเดตเอกสารกำกับข้อมูลจากแหล่งข้อมูลต่างๆ ใน Data Model หรือแม้กระทั่งใช้สร้างเอกสารเหล่านี้ขึ้นมาถ้ายังไม่มีการจัดทำขึ้นมาก่อน ซึ่งข้อดีที่ Data Profiling ใช้ตรวจสอบความถูกต้องของเอกสารดังกล่าวได้ถือว่าเป็นสำคัญมาก เนื่องจากหลายองค์กรมักจะมีเอกสารที่ล้ำสมัยและไม่สมบูรณ์แบบ

ตัวอย่างเครื่องมือที่ให้ความสำคัญกับ Data Profiling ได้แก่ Quality Manager ของ Informaix/Ardent, Migration Architect ของ Evoke Software Inc. อย่างไรก็ตาม ถึงแม้ว่าซอฟต์แวร์ทั้งสองนี้จะเป็นเครื่องมือในการทำ Data Profiling ทั้งคู่ แต่ก็ยังมีลักษณะที่ค่อนข้างจะแตกต่างกันมาก ตัวอย่างเช่น Migration Architect จะเน้นที่เทคนิคการค้นหาและการวิเคราะห์ข้อมูล (Data Recovery and Analysis) เพื่อที่จะหากฎทางธุรกิจมากกว่า Quality Manager ในขณะที่ Quality Manager จะเน้นที่การตรวจสอบความถูกต้องของข้อมูลมากกว่า

### 6.4 ประโยชน์ของ Data Profiling

1. ช่วยให้เราพบข้อมูลที่อาจทำให้เกิดความผิดพลาดเร็วขึ้น
2. สามารถนำเข้า (Import) ข้อมูลจากหลายๆ แพลตฟอร์ม
3. สามารถใช้วิธีการสุ่มตัวอย่าง (Sampling) ที่หลากหลายในแบบอินเทอร์แอคทีฟได้
4. แสดงผลผ่าน GUI บนแพลตฟอร์มวินโดวส์
5. สามารถแนะนำ Metadata และระบุสิ่งผิดปกติ (Anomaly) ได้
6. ช่วยระบุข้อมูลที่ซ้ำซ้อนกัน
7. ช่วยกำหนดความสัมพันธ์ตามหน้าที่ (Functional Dependency) ของข้อมูล
8. ค้นพบข้อมูลและทำรายงานข้อมูลได้

ผลลัพธ์ที่ได้จากการทำ Data Profiling ไม่เพียงใช้ได้ในการพัฒนา Data Warehouse เท่านั้น

แต่ยังสามารถใช้ในโครงการรวบรวมและโยกย้ายข้อมูลใดๆ ก็ได้ เช่น ใช้ในการย้ายข้อมูลจาก  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบที่มีอยู่ (Legacy System) ไปแอปพลิเคชันสำเร็จรูป เช่น ระบบวางแผนการใช้ทรัพยากรในองค์กร หรือ ERP และระบบบริหารความสัมพันธ์กับลูกค้า หรือ Customer Relationship Management (CRM) นอกจากนี้บางองค์กรยังใช้ประยุกต์ใช้ Data Profiling ในการพัฒนาระบบอีบิสซิเนส และรวมระบบดังกล่าวเข้ากับระบบแบ็กออฟฟิศ (Back-office) ที่มีอยู่อีกด้วย

หลังจากขั้นตอนการทำ Data Profiling ซึ่งช่วยให้เราสามารถระบุปัญหาคุณภาพข้อมูลได้ เราก็จะสามารถใช้เครื่องมือ Data Reengineering เพื่อที่จะตรวจสอบ และแก้ปัญหาที่พบนั้น ซึ่งเครื่องมือเหล่านี้มักจะเน้นที่การดึงข้อมูลจากฟิลด์ข้อความที่อยู่ในรูปแบบใดๆ ก็ได้ จากนั้นก็จะกำจัดหรือรวบรวมข้อมูลเกี่ยวกับชื่อและที่อยู่เสียใหม่ ตัวอย่างเครื่องมือที่สนับสนุนการทำ Data Reengineering ก็คือ Trillium Software System และ Integrity ของ Vality Technology Inc.

กล่าวโดยสรุปแล้ว การใช้เครื่องมือทั้งสามประเภทรวมกัน ไม่ว่าจะเป็นในกระบวนการ Data Profiling, Data Engineering หรือ ETL เป็นเทคนิคในการพัฒนา Data Warehouse โดยอาศัยการค้นหาปัญหาที่อาจเกิดขึ้นเนื่องจากคุณภาพของข้อมูลไม่ดีพอ ก่อนที่จะโยกย้ายข้อมูล เทคนิคนี้เสนอขึ้นมาเพื่อทดแทนวิธีการพัฒนาที่ใช้ ETL เพียงอย่างเดียว เนื่องจากเทคนิคนี้ช่วยให้เราค้นพบและแก้ไขปัญหาด้านคุณภาพของข้อมูลในขั้นแรกๆ ของกระบวนการพัฒนา (Development Cycle) Data Warehouse ซึ่งช่วยลดความเสี่ยงที่เกิดจากคุณภาพของข้อมูลใน Data Warehouse นั้นเอง อย่างไรก็ตาม วิธีดังกล่าวก็ยังไม่ใช่คำตอบของปัญหาทั้งหมด ดังเช่นที่ Larry English ที่ปรึกษาเกี่ยวกับการจัดการคุณภาพข้อมูล กล่าวไว้ว่า "การปรับปรุงคุณภาพข้อมูล เป็นเพียงหนึ่งในการปรับปรุงกระบวนการทำงานในหลายๆ ระบบเพื่อที่จะลดเหตุที่ทำให้เกิดความเสียหายของข้อมูลเท่านั้น" (อริณี ชีวะเกรียงไกร. 2543ข : 128)

## บทที่ 7

### การศึกษาเครื่องมือที่ใช้วิเคราะห์ข้อมูลในคลังข้อมูล

#### Microsoft SQL Server Analysis Services

##### 7.1 บทบาทของ Microsoft SQL Server Analysis Services

Microsoft SQL Server Analysis Services ถูกประกาศว่าเป็นเซิร์ฟเวอร์ฐานข้อมูลแบบหลายโดเมนชั้น (Multidimensional) โดย Analysis Services นั้นจะถูกใช้เพื่อวิเคราะห์ข้อมูลขนาดใหญ่ได้รวดเร็ว ยืดหยุ่นและง่าย ซึ่งเป็นส่วนที่ขาดไม่ได้ในการวางกลยุทธ์ในด้านคลังข้อมูลขององค์กร แต่ Analysis Services ก็ไม่สามารถตอบสนองความต้องการทั้งหมดของคลังข้อมูลได้ เพียงแต่จะมีบทบาทในการช่วยเรียกรับข้อมูลจากคลังข้อมูลให้เป็นไปอย่างรวดเร็ว และทำให้การแบ่งส่วนข้อมูล (Slice) และการตัดทอน (Dice) ข้อมูลทำได้ง่ายขึ้น ซึ่ง Analysis Services ไม่ได้จัดเครื่องมือค้นหาที่ใช้งานได้ง่ายไว้ให้โดยตรง แต่ผู้ใช้สามารถใช้เครื่องมือในการทำรายงานที่เรียกว่า Microsoft Excel 2000 PivotTable ทำหน้าที่เป็นบราวเซอร์แบบง่ายๆ สำหรับคลังข้อมูลของ Analysis Services ได้ (พรพิมล อนันตคามนิช. 2544 : 3)

ส่วนข้อกำหนดความต้องการทำให้ข้อมูลในคลังข้อมูลมีความสม่ำเสมอ มีความน่าเชื่อถือ และสามารถเชื่อถือได้นั้นผู้ใช้จะต้องทำให้สำเร็จด้วยตัวเองก่อนที่จะใช้ Analysis Services ทำงาน หรืออีกนัยหนึ่งคือ Analysis Services จะสันนิษฐานผู้ใช้มีคลังข้อมูลที่สมบูรณ์ถูกต้อง รวมทั้งคลังข้อมูลที่สามารถนำไปใช้งานได้ที่ถูกสร้างโดยใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ซึ่งถึงแม้ว่า Analysis Services จะขายพร้อมกับ SQL Server 2000 ก็ไม่ได้หมายความว่าต้องใช้ SQL Server เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์เท่านั้น ยังสามารถใช้ Analysis Services ร่วมกับระบบจัดการฐานข้อมูลเชิงสัมพันธ์อื่นๆ ได้ไม่ว่าจะเป็น Oracle, DB2, Informix รวมทั้ง Microsoft Access

Analysis Services จะสร้างเลเยอร์ในส่วนบนสุดของคลังข้อมูลเชิงสัมพันธ์ที่มีอยู่แล้วขึ้นมาใหม่ จุดประสงค์ของเลเยอร์ใหม่นี้ก็เพื่อทำให้การเข้าถึงข้อมูลทำได้อย่างรวดเร็วและมีความยืดหยุ่นสูง

## 7.2 การทำงานของ Analysis Services

Analysis Services จะอยู่ระหว่างฐานข้อมูลเชิงสัมพันธ์และแอปพลิเคชันรายงาน หน้าที่ของ Analysis Services คือเป็นเลเยอร์ระหว่างกลางซึ่งแปลงคลังข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปแบบที่จะทำให้สร้างรายงานการวิเคราะห์ได้เร็วและยืดหยุ่น

คอมโพเนนต์พื้นฐานของ Analysis Services คือ Analysis Server ซึ่งจะทำงานเป็นบริการในระบบ Microsoft Windows 2000 หรือ Microsoft Windows NT ซึ่ง Analysis Server จะขยายข้อมูลข่าวสารจากคลังข้อมูลและสร้างสิ่งที่เรียกว่า คิวบ์ (Cube) โดยจำนวนของค่าต่างๆ ในคิวบ์เป็นผลผลิตของจำนวนสมาชิกต่างๆ ในแต่ละโดเมนชั้น จำนวนของหน่วยที่มีอยู่ในคิวบ์ก็คือผลผลิตของจำนวนหน่วยในแต่ละด้านของคิวบ์ อย่างไรก็ตาม ในเชิงวัตถุ คิวบ์จะอ้างถึงอ็อบเจกต์โดยใช้ 3 โดเมนชั้น ใน Analysis Services คิวบ์จะเป็นแหล่งที่เก็บส่วนกลางของค่าต่างๆ ซึ่งเป็นไปได้ที่อาจต้องการรวมค่าในรายงานเฉพาะ คิวบ์สามารถมีโดเมนชั้นได้ตั้งแต่ 1 ถึง 24 โดเมนชั้น (นับ measure เป็นโดเมนชั้นด้วย) โดยโดเมนชั้นจะเป็นเครื่องชี้ถึงแนวทางอิสระที่ค่าต่างๆ ของข้อมูลที่สามารถแบ่งออกเป็นส่วนย่อยๆ ได้

ในการทำงานของ Analysis Services รายงานจะรับข้อมูลจากคิวบ์และคิวบ์จะรับข้อมูลจากคลังข้อมูลอีกทีหนึ่ง คิวบ์เป็นวิธีในการทำแพ็คเกจ หรือการเสนอค่าต่างๆ จากคลังข้อมูล ส่วน Analysis Server เป็นคอมโพเนนต์ที่ใช้สร้างคิวบ์ และในขั้นตอนสุดท้ายคิวบ์จะให้ค่าต่างๆ ไปยังแอปพลิเคชันของไคลเอนต์

- **Analysis Services Administrator Tools**

เมื่อติดต่อกับ Analysis Server ในฐานะเป็นผู้ดูแล (Administrator) เราจะใช้แอปพลิเคชัน Analysis Manager แอปพลิเคชันนี้จะยอมให้มีการกำหนดและบรรจุคิวบ์ได้ โปรแกรม Analysis Manager จะสื่อสารกับ Analysis Server ด้วยการใช้เลเยอร์ที่เรียกว่า Decision Support Object (DSO) ซึ่ง DSO นี้จะเป็นชุดคำสั่งในการเขียนโปรแกรมที่ยอมให้แอปพลิเคชันควบคุม Analysis Server ได้ ส่วน Analysis Manager เป็นแอปพลิเคชันง่ายๆ ที่ไมโครซอฟท์สร้างขึ้นซึ่งใช้ DSO ควบคุมการทำงานของ Analysis Services ซึ่งเราสามารถใช่ DSO ในการเขียนแอปพลิเคชันขึ้นเองเพื่อจัดการกับ Analysis Server ก็ได้และในความเป็นจริงสิ่งนี้ก็ในรูปแบบของ Analysis Server ที่สามารถใช้งานได้ผ่าน DSO แต่ไม่ได้ถูกรวมไว้ในแอปพลิเคชันของ Analysis Manager

- **Analysis Services User Tools**

เมื่อติดต่อกับ Analysis Server ในฐานะเป็นผู้ใช้ เราต้องทำการร้องขอเพื่อรับค่าจากคิวบ์เข้าไปในรายงานด้วยการใช้บราวเซอร์ของแอปพลิเคชันไคลเอนต์ เครื่องมือของ Analysis Manager จะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกใช้โดยผู้ดูแล ซึ่งมีเครื่องมือบราวเซอร์อย่างง่าย ๆ รวมอยู่ภายในด้วย Microsoft Office 2000 จะรวม PivotTable Report ไว้ใน Excel และ PivotTable List ActiveX Control เป็นส่วนหนึ่งของ Office Web Component เครื่องมือทั้งสองประเภทนี้เป็นความสามารถของบราวเซอร์ที่สามารถเรียกดูคิวบ์ของ Analysis Services ได้ นอกจากนี้ Analysis Services ยังรวมโปรแกรมที่เรียกว่า MDX Sample ซึ่งยอมให้สอบถามหรือคิวรีคิวบ์ด้วยการใช้กลุ่มคำสั่ง MDX (Multidimensional Expression) ได้

แอปพลิเคชันของไคลเอนต์ต่างๆ ตัวจะต้องติดต่อสื่อสารกับ Analysis Server ผ่านทางขั้นตอนการทำงานของไคลเอนต์ที่เรียกว่า PivotTable Service ซึ่งต้องอยู่ในเครื่องคอมพิวเตอร์ของไคลเอนต์เอง PivotTable Service จะถูกติดตั้งโดยอัตโนมัติบนเครื่องคอมพิวเตอร์เมื่อทำการติดตั้ง Analysis Services หรือเมื่อติดตั้ง Office 2000 โดยที่ PivotTable Service สามารถทำงานในระบบปฏิบัติการ Microsoft Windows เวอร์ชันใดก็ได้ตั้งแต่ 95 เป็นต้นไป และยังสามารถเรียกรับค่าจากคิวบ์โดยไม่ต้องใช้ PivotTable Service ในการติดต่อสื่อสารกับ Analysis Server ได้ด้วย แต่ในความเป็นจริงแม้ว่าจะใช้ Analysis Manager เพื่อจัดการคิวบ์และใช้บราวเซอร์เพื่อเรียกดูคิวบ์นั้น Analysis Manager ก็จะสื่อสารกับ PivotTable Service และ PivotTable Service จะเรียกรับข้อมูลจาก Analysis Server อีกทอดหนึ่ง

PivotTable Service จะซ้อนกับการทำงานของ Analysis Server หลายอย่าง เราสามารถใช้ PivotTable Service ในการสร้าง โหลดและเรียกดูโลคอลคิวบ์บนเครื่องไคลเอนต์ได้แม้ว่าจะไม่มีการเชื่อมต่อกับเซิร์ฟเวอร์ก็ตาม

## บทที่ 8

### กรณีศึกษาการพัฒนาระบบคลังข้อมูลสำหรับวิเคราะห์การขาย

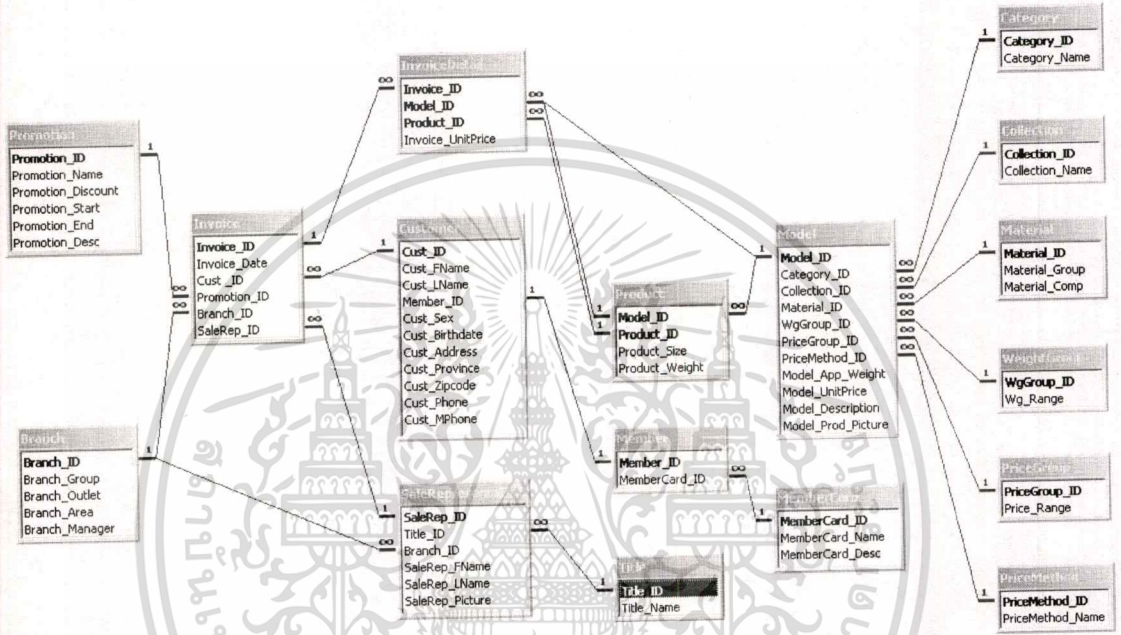
เนื่องจากระบบงานขายสินค้าเป็นระบบงานที่มีลักษณะการประมวลผลเป็นประจำวัน ข้อมูลที่ผ่านการประมวลผลในระบบทั้งหมดจะถูกจัดเก็บลงบนฐานข้อมูลในรูปแบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งปริมาณข้อมูลนับวันก็จะมีจำนวนมากขึ้น เมื่อผู้บริหารต้องการทราบข้อมูลสรุปในรูปแบบที่เป็นรายงาน และสถิติต่างๆ เพื่อนำข้อมูลไปใช้ช่วยในการตัดสินใจในการบริหารการขาย ในการวางแผนการผลิต การควบคุมสินค้าคงเหลือ ตลอดจนวางแผนการออกแบบและพัฒนาผลิตภัณฑ์ การใช้เวลาในการสืบค้นข้อมูลทีนานและในบางครั้งก็ไม่ทันต่อความต้องการที่จะใช้ข้อมูลของผู้บริหารทำให้เกิดปัญหาเรื่องการควบคุมสินค้าคงเหลือ เนื่องจากเครื่องประดับทองคำนับเป็นสินค้าที่มีมูลค่าสูง และทองคำแท่งซึ่งเป็นวัตถุดิบที่ใช้ในการผลิตมีราคาที่สูงจนอยู่ตลอดเวลา จึงมีความเสี่ยงสูงในการซื้อทองคำเพื่อกักตุนไว้ หากมีการวางแผนการผลิตสินค้าที่ไม่สอดคล้องกับการขายและความต้องการของตลาดจึงเกิดผลกระทบต่อบริษัทค่อนข้างมาก จากปัญหาดังกล่าวจึงได้เกิดแนวคิดที่จะจัดทำระบบคลังข้อมูลสำหรับงานขายขึ้น โดยเนื้อหาในบทนี้จะได้กล่าวถึงการพัฒนาระบบคลังข้อมูลขึ้นใช้ในบริษัทโกลด์มาสเตอร์ จำกัด (มหาชน) ซึ่งดำเนินธุรกิจการผลิตและจำหน่ายสินค้าเครื่องประดับทองคำ 99.99% ซึ่งมีลักษณะการขายแบบขายปลีกถึงลูกค้าโดยตรงผ่านจุดขายซึ่งอยู่ตามห้างสรรพสินค้าต่างๆ

#### 8.1 ขั้นตอนการดำเนินงาน

1. ศึกษารายละเอียด และโครงสร้างฐานข้อมูลประจำวันของการขายสินค้าที่มีอยู่เดิม
2. วิเคราะห์และออกแบบฐานข้อมูลสำหรับระบบคลังข้อมูล
3. พัฒนาระบบคลังข้อมูลการขายสินค้า
4. การออกแบบและสร้าง Model
5. การวิเคราะห์ข้อมูลและออกรายงานสรุป

### 8.2 ศึกษารายละเอียด และโครงสร้างฐานข้อมูลประจำวันการขายสินค้าที่มีอยู่เดิม

จากฐานข้อมูลการขายสินค้าซึ่งพัฒนาขึ้นโดยโปรแกรม Microsoft Access ซึ่งเป็นฐานข้อมูลประจำวันที่มีอยู่เดิม ข้อมูลการขายสินค้าจะถูกจัดเก็บกระจายอยู่ตามตารางต่างๆ หลายตารางขึ้นอยู่กับลักษณะ และความสัมพันธ์ของข้อมูลนั้น ดังรูปที่ 8.1



รูปที่ 8.1 ER-Diagram ของฐานข้อมูลประจำวันการขายสินค้า

จากการศึกษาโครงสร้างของฐานข้อมูลที่มีอยู่เดิม พบว่ามีข้อมูลต่างๆ เพียงพอและเหมาะสมสำหรับใช้เป็นแหล่งข้อมูลของคลังข้อมูลการขายอยู่แล้ว

### 8.3 การวิเคราะห์และออกแบบฐานข้อมูลสำหรับคลังข้อมูล

จากการศึกษาความต้องการข้อมูลและสารสนเทศของผู้บริหาร พบว่าข้อมูลที่ผู้บริหารสนใจได้แก่

- ข้อมูลเกี่ยวกับยอดขายสินค้าแต่ละประเภท แต่ละสถานที่ และในแต่ละช่วงเวลา ไม่ว่าจะเป็นจำนวนเงิน, จำนวนชิ้นที่ขาย หรือคิดเทียบเป็นเปอร์เซ็นต์ เพื่อเป็นข้อมูลใช้สำหรับการวางแผนการผลิตและควบคุมสินค้าคงคลังให้เป็นไปอย่างเหมาะสม

- ข้อมูลเกี่ยวกับรูปแบบของสินค้าที่ขายไปในแต่ละช่วงเวลา และแต่ละสถานที่ เพื่อเป็นเอกสารข้อมูลใช้สำหรับวางแผนการกระจายสินค้าให้เหมาะสมในแต่ละสาขา เพื่อให้สอดคล้องกับไม่ว่าการณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พฤติกรรมกรบรโภคของลูกค้าที่แตกต่างกันในแต่ละพื้นที่ ตลอดจนเพื่อใช้เป็นแนวทางในการออกแบบและพัฒนาผลิตภัณฑ์ให้สอดคล้องกับความต้องการของลูกค้า

ในการออกแบบฐานข้อมูลสำหรับคลังข้อมูลระบบงานขายสินค้าได้ใช้เทคนิคการออกแบบแบบ Snowflake Schema ซึ่งจากการพิจารณาข้อมูลต่างๆ ทางธุรกิจเกี่ยวกับการขายที่เก็บในฐานข้อมูลเชิงสัมพันธ์เดิมที่มีอยู่ นำมาใช้ในการพิจารณาออกแบบโครงสร้างของคลังข้อมูล ได้ดังนี้

- Fact table เป็นตารางหลักซึ่งใช้จัดเก็บข้อมูลที่เป็นข้อเท็จจริงที่เกิดขึ้นจากการขายสินค้า โดยเก็บข้อมูลการขายเป็น 2 ตาราง คือ

- SalesFact : เก็บข้อมูลที่เกี่ยวข้องกับการขายสินค้า โดยมี Invoice\_Date, Model\_ID, Branch\_ID, Promotion\_ID และ SaleRep\_ID เป็น Foreign keys เพื่อเชื่อมโยงไปยังตารางใดเมนชัน และมี Weight, Units และ Amount เป็น Measure หรือค่าที่ใช้วัด ซึ่งได้จากการเข้ามาไหลเซชันข้อมูลมาอีกทีหนึ่ง

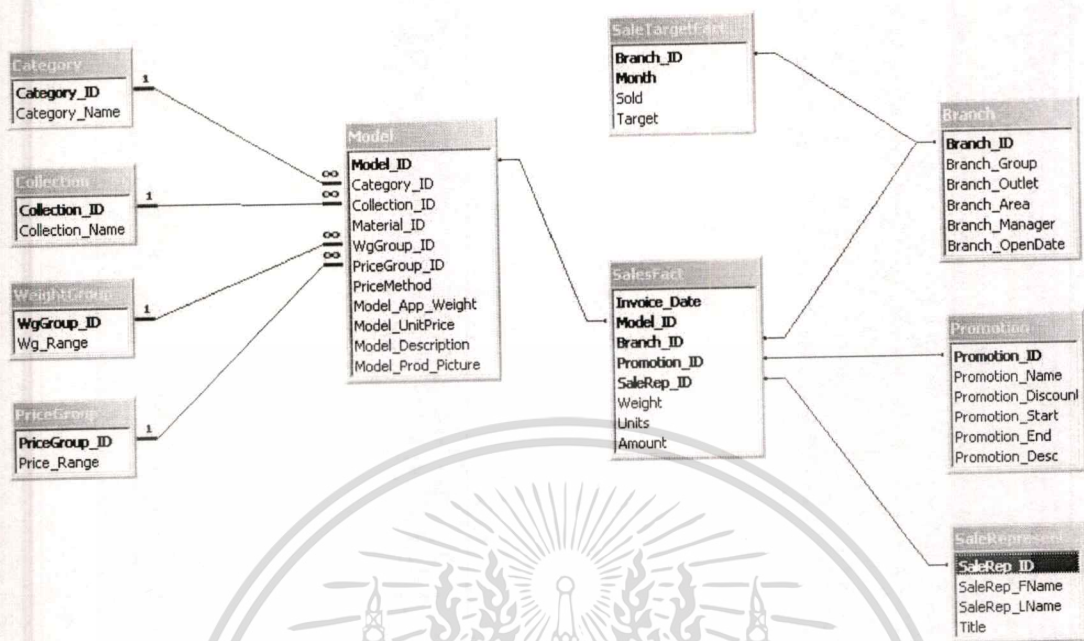
- SaleTargetFact : เก็บข้อมูลการขายสินค้า และเป้าที่กำหนดไว้ในการขายสินค้าของแต่ละสาขา ของแต่ละเดือน โดยมี Branch\_ID และ Month เป็น Foreign keys เพื่อเชื่อมโยงไปยังตารางใดเมนชัน และมี Sold, Target เป็น Measure

- Dimension table ซึ่งเป็นตารางที่เก็บข้อมูลเกี่ยวกับคุณลักษณะต่างๆ ของข้อมูลในตารางหลัก ประกอบด้วยตารางต่างๆ ต่อไปนี้

- ตาราง Branch : เก็บรายละเอียดข้อมูลเกี่ยวกับของจุดขายหรือสถานที่ขายสินค้า
- ตาราง Promotion : เก็บรายละเอียดข้อมูลเกี่ยวกับ โปรแกรมสนับสนุนการขาย
- ตาราง Model : เก็บรายละเอียดข้อมูลเกี่ยวกับสินค้า
- ตาราง SaleRepresentative : เก็บรายละเอียดข้อมูลเกี่ยวกับพนักงานขาย

ซึ่งจากตาราง Model จะมีการเก็บข้อมูลที่ซ้ำๆ กันจำนวนมากจึงได้ทำการ Normalized ตารางดังกล่าว ออกเป็นตาราง Category, Collection, WeightGroup และ PriceGroup เพื่อลดความซ้ำซ้อน ดังจะเห็นได้จากความสัมพันธ์ระหว่างตารางหลักและตารางใดเมนชันใน Snowflake Schema ดังรูปที่ 8.2

สำหรับตารางซึ่งเก็บข้อมูลเกี่ยวกับเวลา จะถูกสร้างเป็นใดเมนชันขึ้นภายหลังขณะสร้างคิวบ์ โดยโปรแกรม Analysis Services โดยสร้างจาก Invoice\_Date และ Month จากตาราง SalesFact และ SaleTargetFact

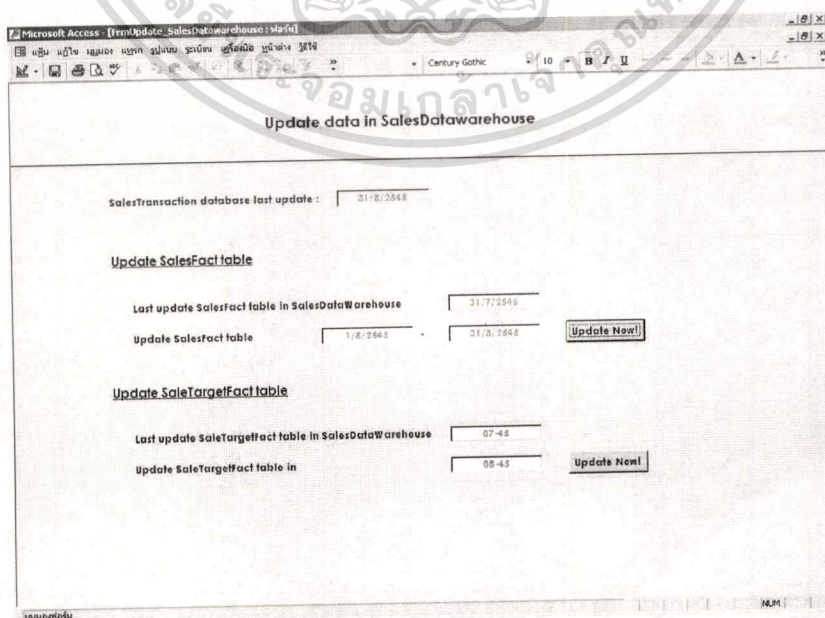


รูปที่ 8.2 Snowflake Schema for Sales

8.4 การอัปเดตข้อมูลในคลังข้อมูล

สำหรับการอัปเดตข้อมูลในคลังข้อมูล SalesDatawarehouse ให้ตรงกับฐานข้อมูลประจำวัน นั้นจะมีการอัปเดตข้อมูลเป็นประจำเดือนละหนึ่งครั้ง โดยผู้ดูแลระบบ ซึ่งมีขั้นตอนดังนี้

- เปิดฟอร์มชื่อว่า FrmUpdate\_SalesDatawarehouse จะปรากฏหน้าจอดังรูปที่ 8.3

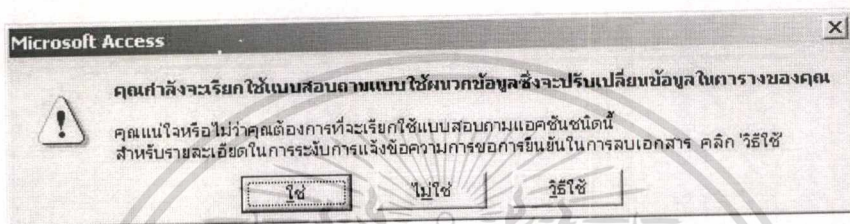


รูปที่ 8.3 หน้าจอแสดงฟอร์มการอัปเดตข้อมูลในคลังข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากฟอร์มดังกล่าวจะแสดงผลการอัปเดตข้อมูลครั้งสุดท้ายทั้งในฐานข้อมูลประจำวัน  
ชื่อว่า SalesTransaction และในคลังข้อมูล SalesDatawarehouse ตลอดจนแสดงช่วงเวลาของข้อมูล  
ใหม่ที่จะทำการเพิ่มเข้าไป

- คลิกปุ่ม Update Now! เมื่อต้องการอัปเดตข้อมูล จะปรากฏหน้าจอให้ยืนยันการเรียกใช้  
แบบสอบถามแบบผนวกข้อมูล (Append Query) และยืนยันจำนวนแถวที่จะทำการผนวกเข้าไปใน  
คลังข้อมูล ดังรูปที่ 8.4 -8.5

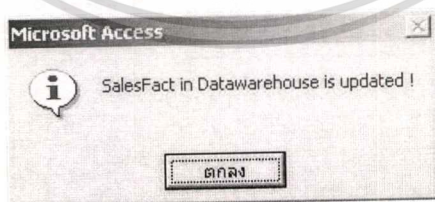


รูปที่ 8.4 หน้าจอแสดงการเรียกใช้แบบสอบถามแบบผนวกข้อมูล



รูปที่ 8.5 หน้าจอแสดงการยืนยันจำนวนแถวที่จะทำการผนวกข้อมูล

- หลังจากยืนยันการผนวกข้อมูลแล้วจะปรากฏหน้าจอแสดงข้อความว่าได้ทำการอัปเดต  
ข้อมูลในคลังข้อมูลเสร็จเรียบร้อยแล้ว ดังรูปที่ 8.6



รูปที่ 8.6 หน้าจอแสดงผลการผนวกข้อมูลที่สำเร็จแล้ว

การอัปเดตข้อมูลหรือการผนวกข้อมูลใหม่เข้าไปในคลังข้อมูลจะเกิดขึ้นเฉพาะในตารางหลัก  
คือตาราง SalesFact และ SaleTargetFact เพียงสองตารางเท่านั้น ส่วนตารางอื่นๆ ซึ่งการอัปเดต  
ข้อมูลเกิดขึ้นไม่บ่อยนักจึงได้ออกแบบให้มีการเชื่อมโยงข้อมูลมาจากฐานข้อมูลประจำวัน เมื่อใดที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการอัปเดตข้อมูลในฐานข้อมูลประจำวัน SalesTransaction ข้อมูลในตารางดังกล่าวในคลังข้อมูล SalesDatawarehouse ก็จะถูกอัปเดตตามไปด้วย

## 8.5 การสร้างและออกแบบ Model

ในกรณีศึกษาที่เราได้ใช้โปรแกรม Microsoft SQL Server 2000 Analysis Services เป็นเครื่องมือที่ใช้สร้างและออกแบบ Model โดยคิวบ์ (Cube) ที่ถูกสร้างขึ้นจาก Analysis Services ก็คือ Model ของเรานั้นเอง การออกแบบคิวบ์มีขั้นตอนดังต่อไปนี้

### 8.5.1 เตรียมการสร้างคิวบ์

- การตรวจสอบโครงสร้างของคลังข้อมูล

คิวบ์ที่จะสร้างขึ้นนี้สร้างจากคลังข้อมูลการขายของฐานข้อมูลชื่อว่า SalesDatawarehouse ซึ่งพัฒนาด้วย Microsoft Access จาก Schema ดังรูปที่ 8.2 แสดงให้เห็นถึงโครงสร้างของฐานข้อมูล ซึ่งประกอบขึ้นจากตารางหลักชื่อว่า SalesFact และ SaleTargetFact ซึ่งตาราง SalesFact ประกอบด้วยคีย์ของไคเมนชัน 5 ตัว คือ Invoice\_Date, Model\_ID, Branch\_ID, Promotion\_ID, SaleRep\_ID และ measure 3 ตัวคือ Weight, Units, Amount ส่วนตาราง SaleTargetFact จะประกอบด้วยคีย์ของไคเมนชันเพียง 2 ตัว คือ Branch\_ID, Month และ measure 2 ตัวคือ Sold, Target

- Model\_ID เป็นคีย์ที่ถูกใช้ร่วมกันกับตารางไคเมนชันที่เป็นลูกโซ่ของตารางไคเมนชัน (Snowflake Schema) ซึ่งเกิดจากการ Normalized ตาราง Model ดังกล่าว
- Branch\_ID, Promotion\_ID และ SaleRep\_ID เป็นคีย์ที่ถูกใช้ร่วมกันกับตารางไคเมนชันเดี่ยว (Star Schema)
- Invoice\_Date และ Month จะแสดงให้เห็นถึงวิธีการที่ Dimension Wizard สามารถสร้างลำดับขั้นทั้งหมดให้เสร็จสมบูรณ์จากการใช้คอลัมน์วันที่เพียงคอลัมน์เดียว

- การสร้างฐานข้อมูล OLAP ใหม่

ใน Analysis Manager ก่อนที่จะสร้างคิวบ์จะต้องสร้างฐานข้อมูล OLAP ที่จะเก็บคิวบ์ก่อน ในฐานข้อมูล OLAP สามารถบรรจุคิวบ์ได้มากกว่า 1 คิวบ์ แต่คิวบ์จะไม่สามารถอยู่ภายนอกฐานข้อมูลได้ โดยในที่นี้เราได้สร้างฐานข้อมูล OLAP ที่ชื่อว่า Sales ขึ้นมาก่อนเพื่อใช้เก็บคิวบ์ที่จะสร้างขึ้นต่อไป

- การระบุแหล่งข้อมูล (Data Source)

คิวบ์ต้องเรียกเก็บค่าของข้อมูลมาจากตารางหลักในคลังข้อมูลเชิงสัมพันธ์ คลังข้อมูลนี้จะถูกกำหนดให้อยู่ภายในฐานข้อมูล OLAP เพื่อเป็นแหล่งข้อมูล คิวบ์ที่แตกต่างกันภายในฐานข้อมูล

OLAP สามารถเรียกรับค่าจากคลังข้อมูลที่ต่างกัน ได้ ดังนั้นฐานข้อมูล OLAP สามารถมีแหล่งข้อมูลได้หลายชุด แต่ว่าแต่ละคิวบ์จะใช้แหล่งข้อมูลได้ที่ละชุดเท่านั้น

แหล่งข้อมูลจะถูกนำไปใช้โดย OLE DB Provider เท่านั้น ซึ่ง OLE DB เป็นเทคโนโลยีมาตรฐานทางอุตสาหกรรมที่ถูกนำมาใช้แทนมาตรฐาน ODBC (Open Database Connectivity) ระบบฐานข้อมูลเชิงสัมพันธ์บางระบบ เช่น Microsoft Access หรือ SQL Server ก็จะใช้ OLE DB Provider เป็นพื้นฐานของการทำงานด้วยเช่นกัน เนื่องจากคลังข้อมูล SalesDatawarehouse เป็นฐานข้อมูลของ Microsoft Access จึงสามารถใช้ Access (Jet) OLE DB Provider ได้โดยการเลือก Data Link Properties เป็น Microsoft Jet 4.0 OLE DB Provider และเลือกแหล่งข้อมูลที่ต้องการติดต่อ

เมื่อทำการเชื่อมต่อแหล่งข้อมูลสำหรับฐานข้อมูล OLAP เรียบร้อยแล้ว เราสามารถใช้แหล่งข้อมูลนั้นในการสร้างไคเมนชันหรือคิวบ์ได้ในลำดับต่อไป สำหรับทุกครั้งที่มีการอัปเดตข้อมูลในคลังข้อมูลก็ต้องทำการรีเฟรชแหล่งข้อมูลใหม่ทุกครั้งด้วยเช่นกัน

### 8.5.2 การออกแบบไคเมนชันและคิวบ์

เมื่อสร้างฐานข้อมูล OLAP และมีแหล่งข้อมูลอย่างน้อยหนึ่งชุดในฐานข้อมูลแล้ว จากนั้นก็สามารถทำการสร้างไคเมนชันและคิวบ์ได้ ไคเมนชันจะมีประโยชน์เมื่อถูกรวมเข้าไปในคิวบ์เท่านั้น ซึ่งคิวบ์จะใช้ลำดับชั้นของไคเมนชันในการสรุป Measure จากตารางหลัก ซึ่งตารางหลักจะเป็นตัวกำหนดเนื้อหาของคิวบ์นั้น และ Measure เป็นจำนวนค่าที่ถูกรวมอยู่ในคิวบ์ หรืออาจกล่าวได้ว่าคิวบ์ต้องมีไคเมนชันอย่างน้อยหนึ่งตัว, Measure อย่างน้อยหนึ่งตัว และตารางหลักหนึ่งตาราง (ตารางเดียวเท่านั้น) เสมอ

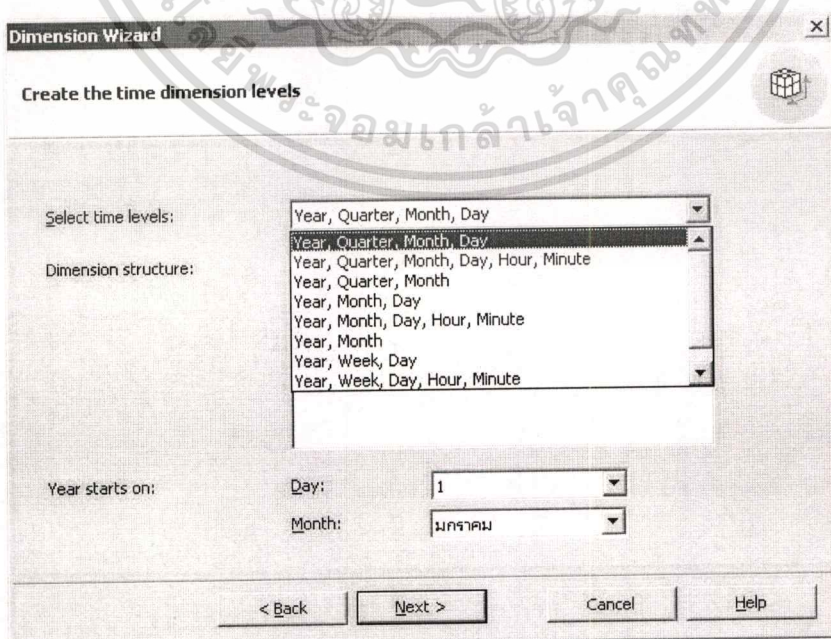
จากฐานข้อมูล OLAP ชื่อว่า Sales ที่ได้ทำการสร้างไว้แล้วนั้น เราได้ทำการสร้างคิวบ์และไคเมนชันไว้ ดังนี้คือ

- ไคเมนชัน Branch
  - สร้างจากราย Branch
  - ประกอบด้วย 2 ลำดับชั้น คือ Branch Group และ Branch Outlet
- ไคเมนชัน SalesPerson
  - สร้างจากราย SaleRepresentative
  - ประกอบด้วย 2 ลำดับชั้น คือ Title และ SaleRep FName
- ไคเมนชัน Category
  - สร้างจากราย Model, Category, PriceGroup
  - ประกอบด้วย 3 ลำดับชั้น คือ Category Name, PriceRange และ Model ID

เอกสารนี้เป็นเอกสารไคเมนชัน Collection ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างจากตาราง Model, Collection, WeightGroup
- ประกอบด้วย 3 ลำดับชั้น คือ Collection Name, WeightRange และ Model ID
- ไคเมนชัน Promotion
  - สร้างจากตาราง Promotion
  - ประกอบด้วย 2 ลำดับชั้น คือ Promotion Name และ Promotion ID
- ไคเมนชัน Time
  - สร้างจากคอลัมน์ Invoice\_Date ในตาราง SalesFact
  - ประกอบด้วย 3 ลำดับชั้น คือ Year, Quarter และ Month
- ไคเมนชัน TimeTarget
  - สร้างจากคอลัมน์ Month ในตาราง SaleTargetFact
  - ประกอบด้วย 3 ลำดับชั้น คือ Year, Quarter และ Month

ในคลังข้อมูลของฐานข้อมูล SalesDatawarehouse จะไม่มีการแยกตารางไคเมนชันสำหรับไคเมนชัน Time หรือ TimeTarget ซึ่งจะมีเพียงคอลัมน์เดียวคือ Invoice\_Date หรือ Month อยู่ในตารางหลักเท่านั้น เราสามารถสร้างไคเมนชันโดยไม่ต้องแยกตารางไคเมนชันออกจากกันเลย โดย Analysis Services สามารถใช้ตัวของตารางจริงเป็นตารางไคเมนชันได้ ซึ่ง Dimension Wizard สามารถกำหนดลำดับชั้นจนเสร็จสมบูรณ์ได้จากคอลัมน์ Date/Time เพียงคอลัมน์เดียว Wizard จะเสนอการกำหนดลำดับชั้นวันที่ที่สัมพันธ์กันหลายๆ แบบให้เลือกตามที่ต้องการ ดังรูปที่ 8.7



รูปที่ 8.7 การเลือกลำดับชั้นของวันที่โดย Dimension Wizard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโดเมนชั้นที่สร้างขึ้นทั้ง 7 โดเมนชั้น นำมาใช้ในการสร้างคิวบ์ ดังนี้

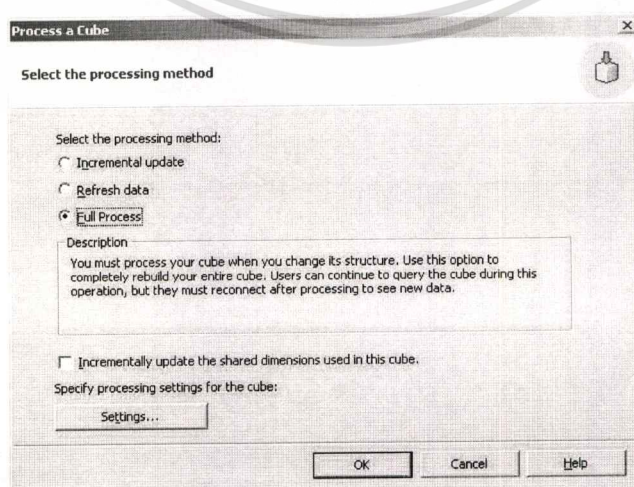
- คิวบ์ Sales and Product : ประกอบด้วย
  - ตารางหลัก 1 ตาราง คือ SalesFact
  - Measure 3 ตัว คือ Weight, Units และ Amount
  - โดเมนชั้น 6 ตัว คือ Time, SalesPerson, Category, Collection, Promotion และ Time
- คิวบ์ Sales and Target : ประกอบด้วย
  - ตารางหลัก 1 ตาราง คือ SaleTargetFact
  - Measure 2 ตัว คือ Sold และ Target
  - โดเมนชั้น 2 ตัว คือ Branch และ TimeTarget
  - สมาชิกที่เกิดจากการคำนวณ 1 ตัว คือ %Differential

(ดูรายละเอียดการสร้างโดเมนชั้น การสร้างคิวบ์ และรายละเอียดของ Metadata ได้ในภาคผนวก)

### 8.5.3 การประมวลผลคิวบ์

เมื่อทำการประมวลผลคิวบ์ Analysis Server จะประมวลผลโดเมนชั้นต่างๆ ซึ่งเคยถูกใช้ในคิวบ์โดยอัตโนมัติ ถึงแม้ว่าเรายังไม่ได้เรียกเข้าไปในโดเมนชั้นนั้นก็ตาม ดังนั้นในการประมวลผลคิวบ์และโดเมนชั้นทั้งหมดของคิวบ์ สามารถจัดการได้ด้วยการใช้คำสั่งเพียงคำสั่งเดียวเท่านั้น โดยการ

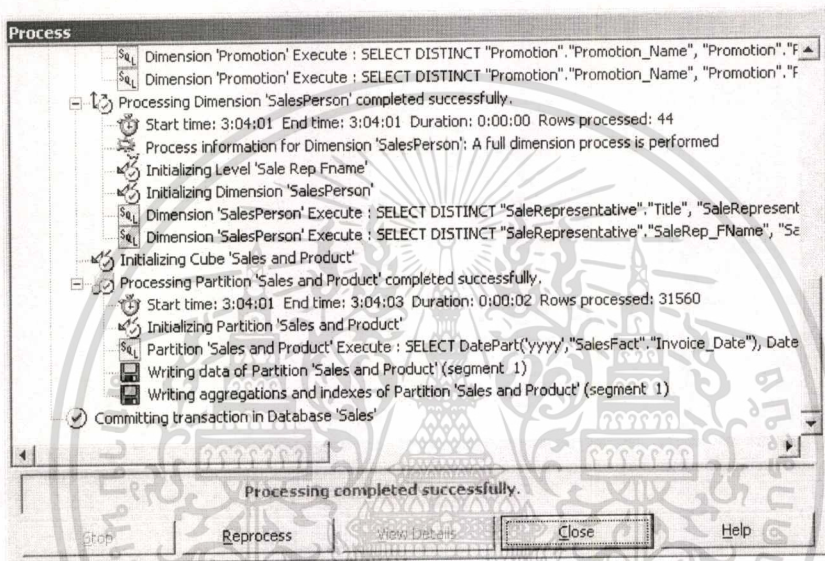
1. คลิกขวาที่คิวบ์ที่ต้องการประมวลผล แล้วคลิก Process กรอบโต้ตอบ Process a Cube จะแสดงให้เห็นตัวเลือกที่มีอยู่เพื่อให้ประมวลผลคิวบ์ได้อย่างสมบูรณ์ ดังรูปที่ 8.8



รูปที่ 8.8 หน้าจอแสดงวิธีเลือกการประมวลผลคิวบ์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากนั้นคลิก OK เพื่อเริ่มต้นการประมวลผล จะปรากฏล็อก (Log) หรือบันทึกที่แสดงรายละเอียดการประมวลผล ซึ่งแสดงรายละเอียดของแต่ละไคเมนชันที่ถูกใช้งาน เมื่อมีการใช้งานไคเมนชันก็จะถูกบันทึกไว้ในล็อกนี้ และเมื่อคิวบ์ถูกประมวลผลก็จะปรากฏอยู่ในอีกกลุ่มหนึ่งของล็อกนี้เช่นกัน ดังรูปที่ 8.9 ซึ่งเมื่อมีการประมวลผลคิวบ์ Analysis Services จะทำการคัดลอกรายละเอียดของค่าต่างๆ จากตารางหลักเข้าไปในโครงสร้างข้อมูลของตัวเอง แต่จะไม่ทำการเก็บค่ารวมหรือการสรุปค่าใดๆ ทั้งสิ้น



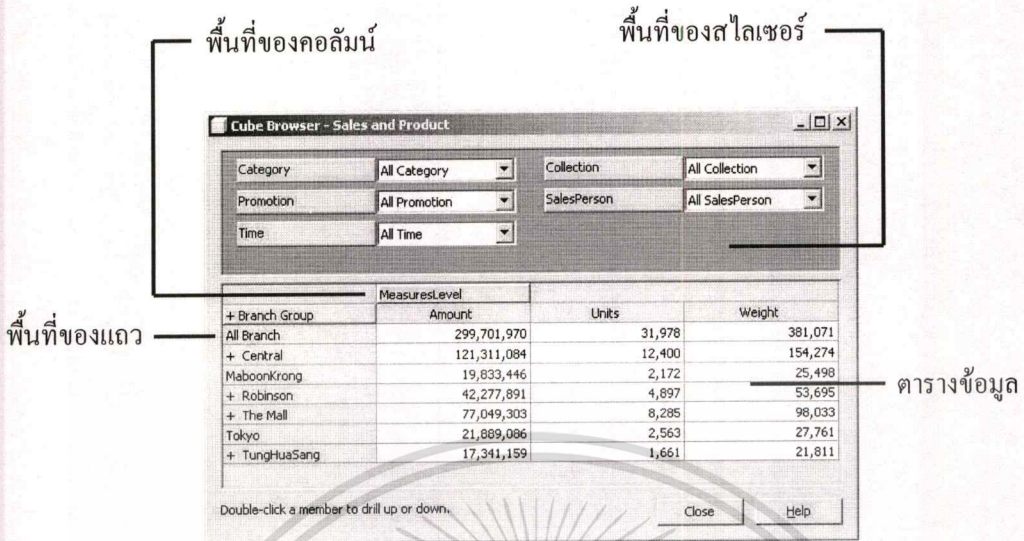
รูปที่ 8.9 หน้าจอแสดงรายละเอียดการประมวลผลคิวบ์

#### 8.5.4 การเรียกดูข้อมูลของคิวบ์

เราสามารถเรียกดูข้อมูลหรือค่าต่างๆ ในคิวบ์ Sales โดยใช้บราวเซอร์ง่ายๆ ที่ Analysis Manager เตรียมไว้ให้ ในหน้าต่าง Cube Browser จะแสดงตารางข้อมูล (Data Grid) ที่มี measure อยู่ด้านบนสุดในพื้นที่ของคอลัมน์ ส่วนไคเมนชันที่เหลือจะปรากฏอยู่ในส่วนบนสุดของบราวเซอร์ในบริเวณที่เรียกว่าพื้นที่ของสไลเซอร์ (Slicer) หรืออาจเรียกว่าฟิลเตอร์ (Filter) หรือเพจฟิลด์ (Page Field) ดังรูปที่ 8.10

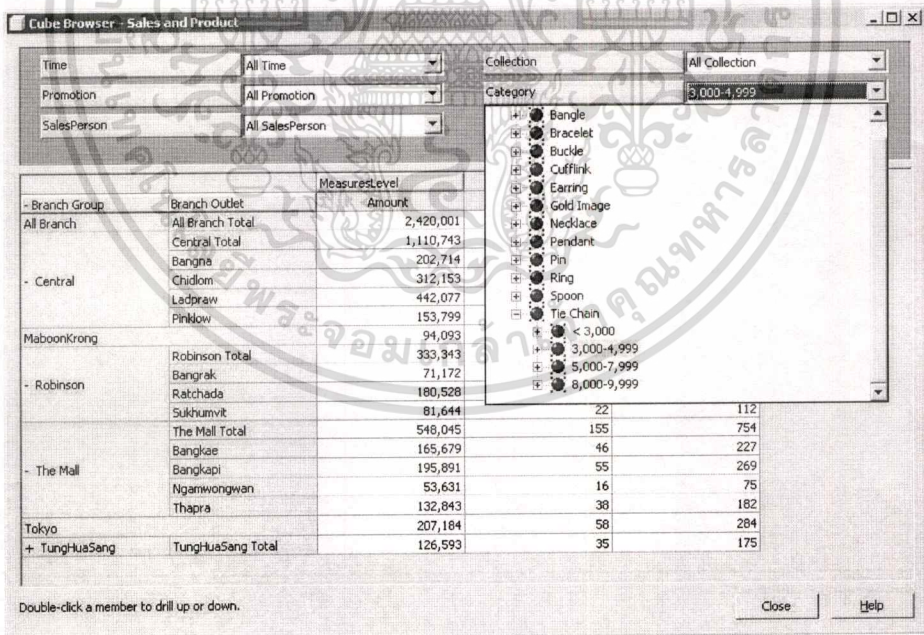
- ในพื้นที่ของแถวหรือพื้นที่ของคอลัมน์จะมีเครื่องหมาย (+) เป็นตัวชี้ให้ทราบว่าสามารถขยายลำดับชั้นหรือ Drill down เพื่อเรียกดูรายละเอียดที่อยู่ในส่วนนี้
- ในพื้นที่ของแถวหรือพื้นที่ของคอลัมน์จะมีเครื่องหมาย (-) เป็นตัวชี้ให้ทราบว่าสามารถย่อลำดับชั้นหรือ Roll up เพื่อดูรายละเอียดที่น้อยที่สุดได้
- บราวเซอร์สามารถทำการสลับตำแหน่งของไคเมนชันได้ หรือหมุนไคเมนชันได้เพื่อ

เลือกดูข้อมูลตามต้องการ โดยการลากปุ่มสลับไปมาระหว่างพื้นที่ของสไลเซอร์ และพื้นที่ของแถว  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.10 หน้าจอแสดง Cube Browser

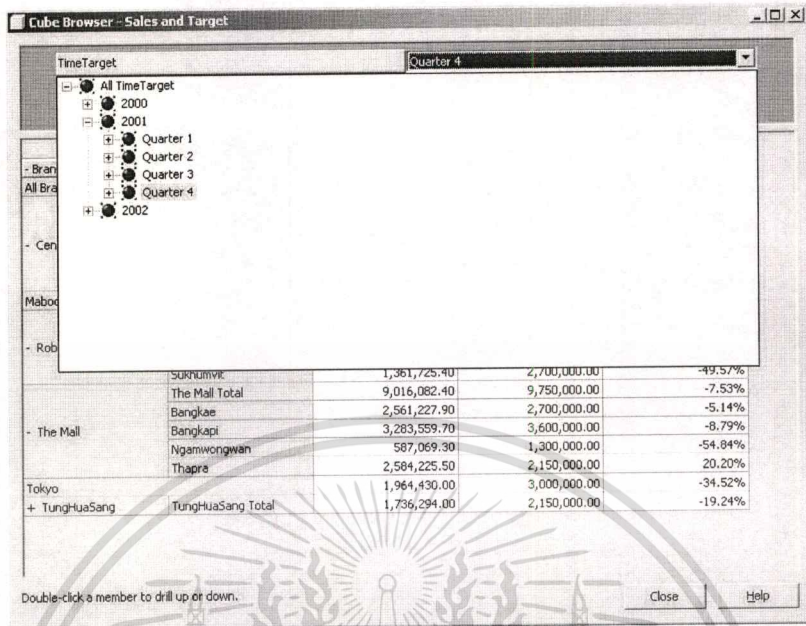
ตัวอย่างที่ 1 : ต้องการดูข้อมูลยอดขายสินค้าประเภทเข็มไทด์ซึ่งราคา 3,000-4,999 บาท  
เปรียบเทียบในแต่ละสาขา



รูปที่ 8.11 Cube Browser แสดงยอดขายสินค้าประเภทเข็มไทด์ราคา 3,000-4,999 บาท

ตัวอย่างที่ 2 : ต้องการดูข้อมูลยอดขายสินค้าเทียบกับเป้าสาขาของแต่ละเดือน ในไตรมาส  
ที่ 4 ปี 2001 เปรียบเทียบทุกสาขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.12 Cube Browser แสดงยอดขายสินค้าเทียบกับเป้า ในไตรมาสที่ 4 ปี 2001

### 8.6 การวิเคราะห์ข้อมูลโดย Excel 2000 PivotTable

รายงานของ Excel PivotTable จะทำให้เราสามารถเลือกและแสดงค่าตัวเลขในตารางแบบไขว้ซึ่งมีลักษณะเหมือนกับที่เราทำงานในคิวบ์แบบ OLAP ในเวอร์ชันก่อนหน้านี้ของ Excel นั้น รายงานของ PivotTable จะดึงค่ามาจากแหล่งข้อมูลเชิงสัมพันธ์เท่านั้น แต่ใน Excel 2000 PivotTable นั้นรายงานของ PivotTable สามารถแสดงข้อมูลจากคิวบ์แบบ OLAP ได้โดยตรงซึ่งจะทำให้การสร้างรายงานและสร้างกราฟต่างๆ ทำได้ง่ายและสะดวก ด้วยการใช้นิวส์แบบ OLAP ให้ค่าต่างๆ กับรายงานของ PivotTable ซึ่งสามารถติดต่อกับ PivotTable Service เพื่อดึงค่าต่างๆ จาก Analysis Server ขึ้นมาใช้ได้

#### 8.6.1 การสร้างการเชื่อมต่อจาก Excel

การสร้างรายงานของ Excel PivotTable ที่อาศัยคิวบ์แบบ OLAP เป็นพื้นฐานนั้นจะต้องใช้ Excel PivotTable Report Wizard ซึ่งวิศวกรครั้งนี้จะใช้แอปพลิเคชัน Microsoft Query เพื่อกำหนดและสร้างไฟล์คิวรีแบบ OLAP ขึ้นมา ไฟล์คิวรีแบบ OLAP จะให้ข้อมูลทั้งหมดที่จำเป็นในการติดต่อกับคิวบ์แบบ OLAP เมื่อ Microsoft Query คืบการควบคุมกลับไป PivotTable Report Wizard แล้ววิศวกรจะใช้ไฟล์คิวรีแบบ OLAP เพื่อเชื่อมต่อกับคิวบ์ต่อไป เมื่อสร้างรายงานของ PivotTable แล้ว Microsoft Query หรือไฟล์คิวรีจะไม่ถูกใช้งานอีกนอกจากเราจะต้องการสร้างรายงานของ PivotTable ขึ้นใหม่เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8.6.2 การเรียกดูคิวบ์ด้วยการใช้รายงานของ PivotTable and PivotChart

รายงานของ Excel PivotTable จะมีลักษณะบางอย่างเหมือน Cube Browser ที่อยู่ใน Analysis Manager แต่รายงานของ PivotTable จะประกอบด้วยพื้นที่ 4 ส่วน คือสามส่วนแรกเป็นพื้นที่ของหน้า (Page Area) พื้นที่ของคอลัมน์ (Column Area) และพื้นที่ของแถว (Row Area) ซึ่งจะประกอบด้วยชื่อสมาชิกและถูกเรียกว่า Axe หรือแกน ในส่วนที่สี่เป็นพื้นที่ของข้อมูล (Data Area) ที่เก็บค่าต่างๆ หรือ Measure ไว้

แถบเครื่องมือของ PivotTable ประกอบด้วยปุ่มแต่ละปุ่มสำหรับแต่ละ Measure รวมทั้งหนึ่งปุ่มสำหรับแต่ละโดเมนชั้นที่ไม่มี Measure ด้วย ปุ่มที่เรียงกันเป็นแถวในแต่ละแถวของแถบเครื่องมือจะประกอบด้วย Measure หรือโดเมนชั้นอย่างใดอย่างหนึ่งเท่านั้น ซึ่งจะไม่มีการประกอบประเภทอยู่พร้อมกัน ไอคอนทางด้านซ้ายของปุ่มที่เรียงเป็นแถวจะเป็นตัวชี้ว่าปุ่มที่อยู่ในแถวเป็น Measure หรือโดเมนชั้น ซึ่งสามารถลากปุ่ม Measure ต่างๆ ไปยังบริเวณพื้นที่ของข้อมูลได้เท่านั้น และสามารถลากปุ่มสำหรับโดเมนชั้นอื่นๆ แต่ต้องไม่ใช่ Measure ไปยังแกนของหน้า คอลัมน์ หรือแถวได้เช่นกัน

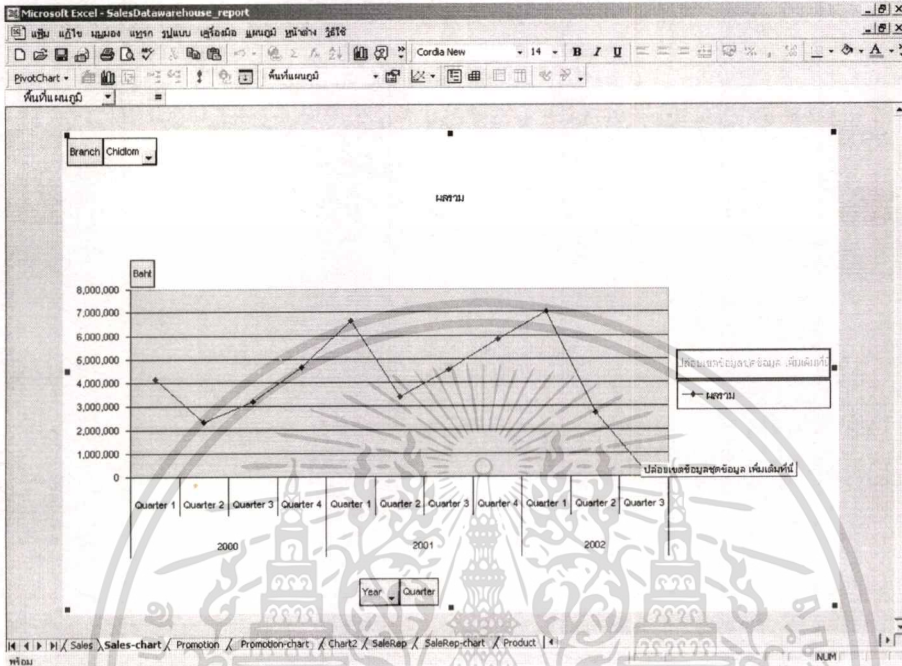
ตัวอย่างที่ 1 : ต้องการดูรายงานยอดขายทั้งหมดเป็นจำนวนชิ้นแยกตามประเภทและแยกตามคอลเลกชันของสินค้า

Category Name	Animalistic	Flowery	Heuristic	Image Impress	Legacy Sense	Modernly	Symbolism	ผลรวมทั้งหมด *
Bangle		219	1,519	5		36		1,779
Bracelet		1,097			755	694	235	2,781
Buckle	15						25	40
Cufflink						8	4	12
Earring	426	305	99		1,333	410	835	3,408
Gold Image				1,093				1,093
Necklace		319	68		1,400	639	781	3,207
Pendant	766	590	2,538		862	862	828	6,446
Pin	79	312			365	64	259	1,079
Ring	1,143	1,220	3,067		1,685	2,551	1,011	10,677
Spoon	55							55
Tie Chain	356	73			275	467	230	1,401
ผลรวมทั้งหมด *	2,440	4,135	7,291	1,093	6,688	5,723	4,208	31,978

รูปที่ 8.13 รายงานยอดขายแยกตามประเภทและคอลเลกชันสินค้าของทุกปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 : ต้องการดูกราฟแสดงแนวโน้มยอดขายโดยแยกเป็น ไตรมาสของทุกปี เฉพาะที่ เซ็นทรัล สาขาชิดลม



รูปที่ 8.14 กราฟแสดงแนวโน้มยอดขายเฉพาะที่ห้างเซ็นทรัล สาขาชิดลม

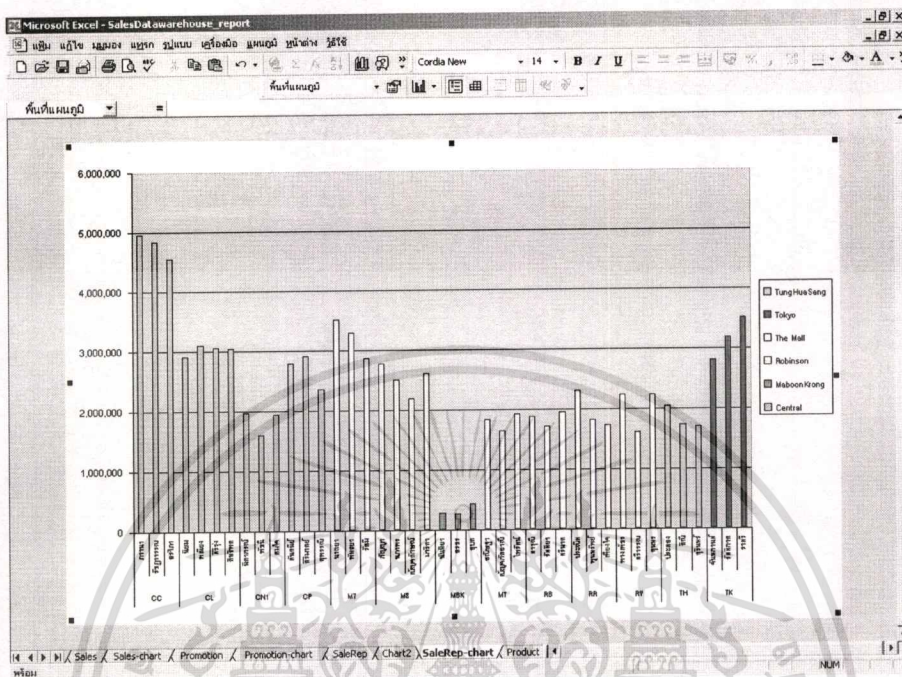
ตัวอย่างที่ 3 : ต้องการดูรายงานยอดขายสินค้าของทุกสาขา โดยเปรียบเทียบเป็นรายเดือน

Year	Quarter	Month	Bangna	Chidlom	Ladprao	Parklow	Central Parkview *	Mahonk Group	Bangkok	Ratchada	Sukhumvit	Robinson HMV HM
2000	Quarter 1	มกราคม	884,440	1,576,249	1,140,007	862,280	4,462,976		527,654	907,782	504,944	1,940,38
		กุมภาพันธ์	584,801	1,591,740	1,356,162	821,816	4,354,519		610,726	580,382	842,036	2,033,14
		มีนาคม	513,671	975,039	1,055,100	625,053	3,168,863		525,097	370,066	400,198	1,255,34
	Quarter 2	เมษายน	449,589	722,661	728,411	479,905	2,380,566		500,545	564,012	509,502	1,574,04
		พฤษภาคม	339,539	895,169	492,214	607,221	2,354,143		290,366	380,859	425,373	1,096,52
		มิถุนายน	124,477	714,765	880,869	489,424	2,209,555		539,760	257,691	506,934	1,304,33
	Quarter 3	กรกฎาคม	486,766	869,596	1,187,475	814,326	3,358,163		346,336	193,892	442,485	980,77
		สิงหาคม	331,626	1,451,492	1,452,313	849,638	4,085,069		382,048	252,188	317,130	951,33
		กันยายน	185,581	889,120	604,732	815,348	2,494,781		415,589	306,469	475,339	1,197,64
	Quarter 4	ตุลาคม	360,681	1,412,920	567,627	713,498	3,054,726	86,787	367,238	627,230	421,491	1,415,93
		พฤศจิกายน	356,862	1,032,784	890,931	304,251	2,584,828	540,932	396,209	390,443	521,261	1,309,93
		ธันวาคม	877,285	2,219,363	1,724,061	662,467	5,483,176	351,597	638,553	1,036,159	712,368	2,587,04
รวม 4 ไตรมาส *			1,594,828	4,665,067	3,182,619	1,680,216	11,122,730	979,311	1,404,000	2,053,832	1,655,120	5,112,92
รวม ปี 2000 *			6,409,210	14,330,890	12,879,912	9,880,177	39,971,340	979,111	8,840,293	8,267,173	6,879,061	17,434,68
2001	Quarter 1	มกราคม	627,905	1,950,508	1,441,038	883,285	4,902,736	705,481	798,564	1,034,194	533,561	2,366,33
		กุมภาพันธ์	678,721	2,886,683	1,229,377	789,711	5,586,492	382,910	606,347	669,079	519,850	1,795,22
		มีนาคม	415,864	1,799,916	814,360	711,423	3,741,563	517,550	490,404	412,256	327,749	1,230,41

รูปที่ 8.15 รายงานยอดขายสินค้าทุกสาขาแยกเป็นรายเดือน

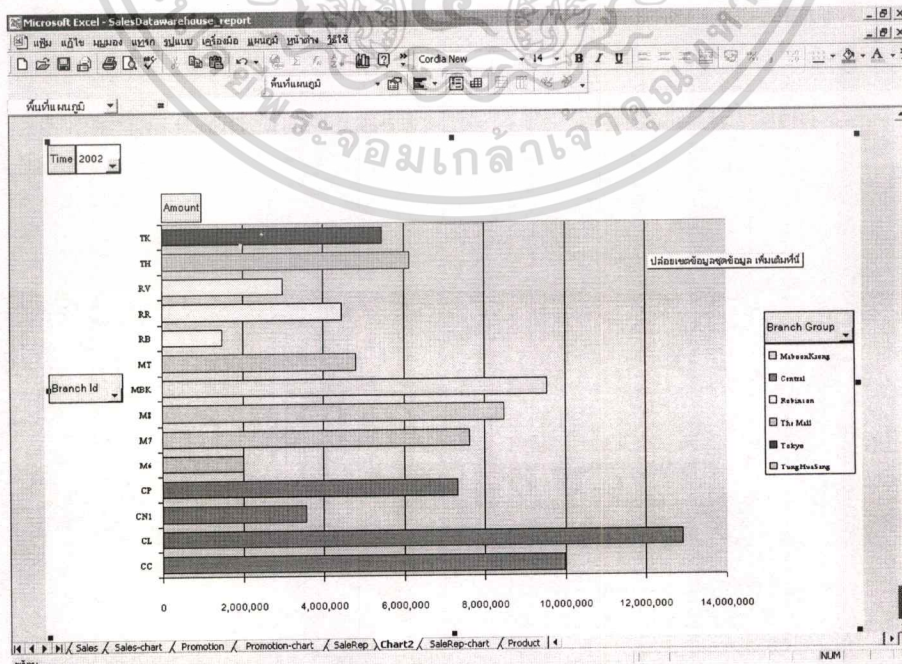
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ การใช้งานเพื่อการค้าโดยไม่ได้รับอนุญาตให้โทษในทางกฎหมาย  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 4 : ต้องการดูกราฟแสดงการเปรียบเทียบยอดขาย ปี 2000 ของพนักงานขายทุกคน



รูปที่ 8.16 กราฟเปรียบเทียบยอดขายปี 2000 ของพนักงานขายทุกคน

ตัวอย่างที่ 5 : ต้องการดูกราฟแสดงการเปรียบเทียบยอดขาย ปี 2001 ของทุกสาขา



รูปที่ 8.17 กราฟแสดงการเปรียบเทียบยอดขาย ปี 2001 ของทุกสาขา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 9

### บทสรุป

#### 9.1 สรุปผลการศึกษา

ระบบคลังข้อมูลการขายสินค้า นับได้ว่าเป็นระบบที่สามารถช่วยแก้ไขปัญหที่เกิดขึ้นในองค์กรได้เป็นอย่างดี ซึ่งการนำโปรแกรม Microsoft SQL Server 2000 Analysis Services มาเป็นเครื่องมือในการสร้างคิวบ์ โดยใช้ร่วมกับ Microsoft Excel 2000 PivotTable ในการวิเคราะห์ข้อมูล ทำให้สามารถตอบคำถาม และแสดงผลข้อมูลเบื้องต้นที่ผู้บริหารต้องการได้ในเวลาที่รวดเร็ว รวมทั้งสามารถเลือกการแสดงผลข้อมูลได้หลายรูปแบบ ซึ่งสามารถเลือกมุมมองของข้อมูลตามใดเมนชั้นที่ต้องการได้ ทำให้การเข้าถึงข้อมูลทำได้อย่างรวดเร็วและมีความยืดหยุ่นสูง ซึ่งนับได้ว่าเป็นพื้นฐานในการที่จะพัฒนาระบบคลังข้อมูลประเภทอื่นๆ สำหรับองค์กรต่อไปในอนาคต

#### 9.2 ข้อเสนอแนะ

ในการที่จะนำเทคโนโลยีคลังข้อมูลมาใช้ในองค์กรนั้น จำเป็นจะต้องมีบุคลากรที่มีความรู้ความสามารถในการออกแบบฐานข้อมูล และเข้าใจระบบงานได้เป็นอย่างดี จึงจะสามารถสร้างฐานข้อมูลที่เหมาะสมกับลักษณะงานขององค์กร โดยจะต้องเป็นฐานข้อมูลซึ่งมีโครงสร้างข้อมูลที่ใช้เข้าใจและใช้งานได้ง่าย ช่วยสนับสนุนการวิเคราะห์ข้อมูลปริมาณมากๆ ได้ ในเวลาที่รวดเร็ว และยังต้องมีการเลือกใช้เครื่องมือวิเคราะห์ข้อมูลที่เหมาะสม ง่ายต่อการใช้และพัฒนาระบบงาน เพื่อช่วยให้การวิเคราะห์ข้อมูลเป็นไปได้อย่างมีประสิทธิภาพ มีความถูกต้อง แม่นยำ ตรงกับความต้องการที่จะใช้ข้อมูลขององค์กร

## บรรณานุกรม

กรองรัตน์ กังวาลไกล. 2544. “Extraction, Transformation และ Load (ETL) ใครว่าทำเองไม่ได้?”  
**PC MAGAZINE**. เมษายน 2544 : 119-122.

พรพิมล อนันควานิซ. 2544. **คัมภีร์นักวิเคราะห์ Microsoft SQL Server 2000 Analysis Services Step by Step**. กรุงเทพฯ : สามย่าน.COM.

สุปรียา หล่อกิตติวิณิชย์. 2544. “Data Mining ขุดทองจากเหมืองข้อมูล.” **PC MAGAZINE**.  
พฤษภาคม 2544 : 175-178.

อนุชิต สุเมธวิทย์. 2541. “Data-Mart เปิดตลาดข้อมูล.” **BYTE THAILAND**. มกราคม 2541 :  
141-146.

อรินิ ชีวะเกรียงไกร. 2543ก. “วิธีพัฒนาลังข้อมูล ตอนที่ 1.” **PC MAGAZINE**. กันยายน 2543 :  
115-118.

อรินิ ชีวะเกรียงไกร. 2543ข. “วิธีพัฒนาลังข้อมูล ตอนจบ.” **PC MAGAZINE**. ตุลาคม 2543 :  
115-118.

Berson, Alex and Smith, Stephen J. 1997. **Data Warehousing, Data Mining, and OLAP**.  
Washington, DC : McGraw-Hill.

Rob, Peter and Coronel, Carlos. 2000. **Database Systems: Design, Implementation, and Management**. 4<sup>th</sup> ed. Cambridge, MA : Course Technology.

Simon, Alan. 2540. “ไคลเอ็นต์คิมิซัยไปกว่าครึ่ง.” **BYTE THAILAND**. มกราคม 2540 : 106-  
109.

Weldon, Jay-Louise and Joch, Alan. 2540. “หนทางสู่ Data Warehouse ที่ดีสำหรับคุณ.” **BYTE THAILAND**. มกราคม 2540 : 103-105.

Weldon, Jay-Louise. 2540. “ก่อสร้างสร้าง Warehouse .” **BYTE THAILAND**. มกราคม 2540 :  
124-129.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก

### ● Metadata ของ Cube : Sales and Product

The screenshot displays the 'Sales and Product' cube metadata in Microsoft Analysis Manager. The left pane shows the tree structure with 'Sales and Product' selected under 'Cubes'. The main pane shows the following details:

**Cube: Sales and Product**

- Dimensions:** Time, Branch, Promotion, SalesPerson, Category, Collection
- Time:** (All), Year, Quarter, Month
- Branch:** (All), Branch Group, Branch Outlet
- Promotion:** (All), Promotion Name, Promotion Id
- SalesPerson:** (All), Title, Sale Rep Fname
- Category:** (All), Category Name, Price Range, Model Id
- Collection:** (All), Collection Name, Wg Range, Model Id
- Measures:** Amount, Units, Weight
- Calculated Members:** None
- Calculated Cells:** None
- Actions:** None
- Named Sets:** None
- Source Tables:** SalesFact, Branch, Category, Model, PriceGroup, Promotion, SaleRepresentative, WeightGroup, Collection
- Fact Table:** SalesFact
- Source Cube:** Not Applicable
- Source Database:** Not Applicable
- Source Server:** Not Applicable
- Processed:** 11/10/2545 4:25:17
- Status:** Processed
- Size:** 5.61MB
- Cube Is:** Read-only

**Partitions**

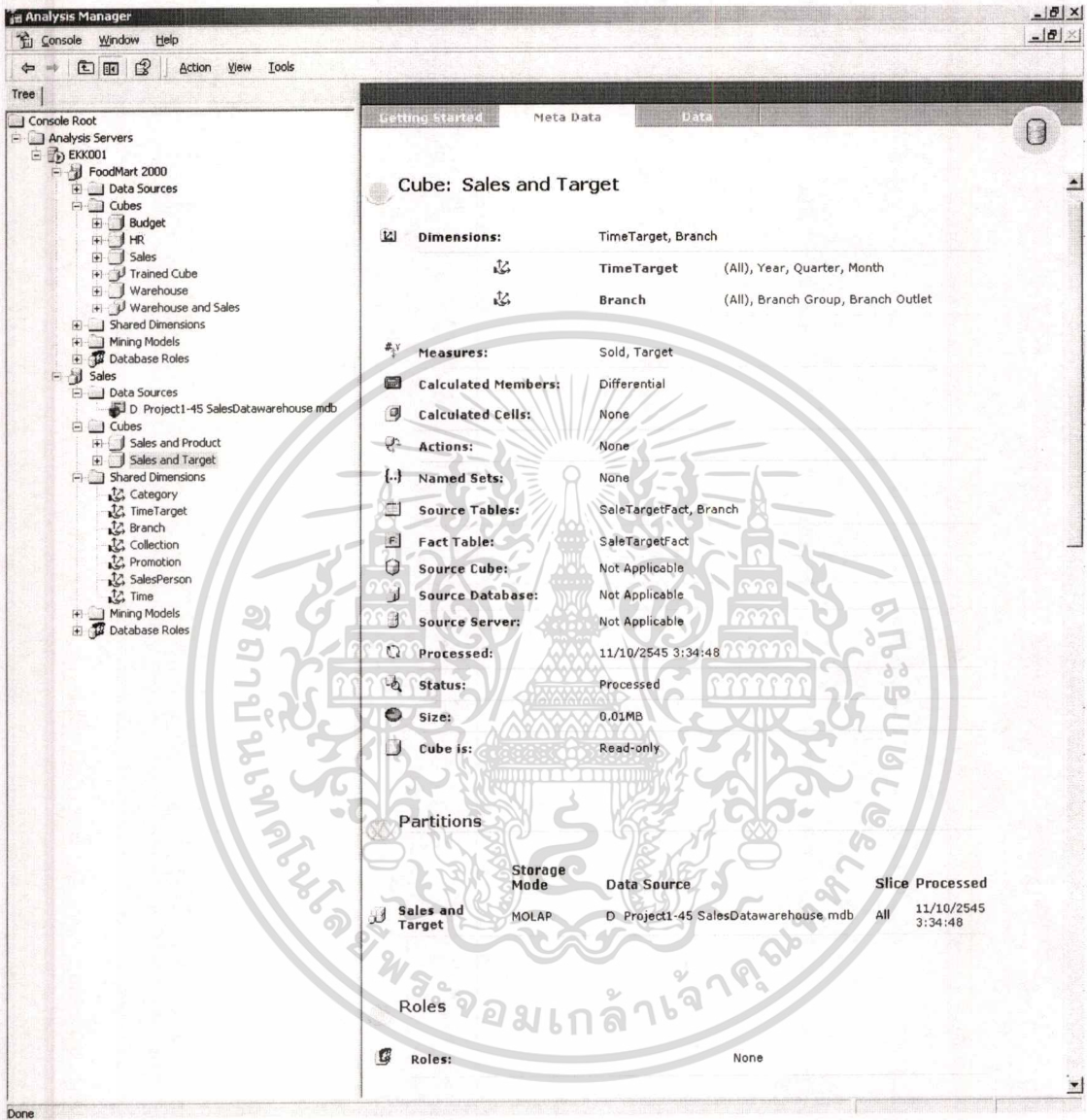
	Storage Mode	Data Source	Slice Processed
Sales and Product	MOLAP	D Project1-45 SalesDatawarehouse mdb	All 11/10/2545 4:25:12

**Roles**

Roles: None

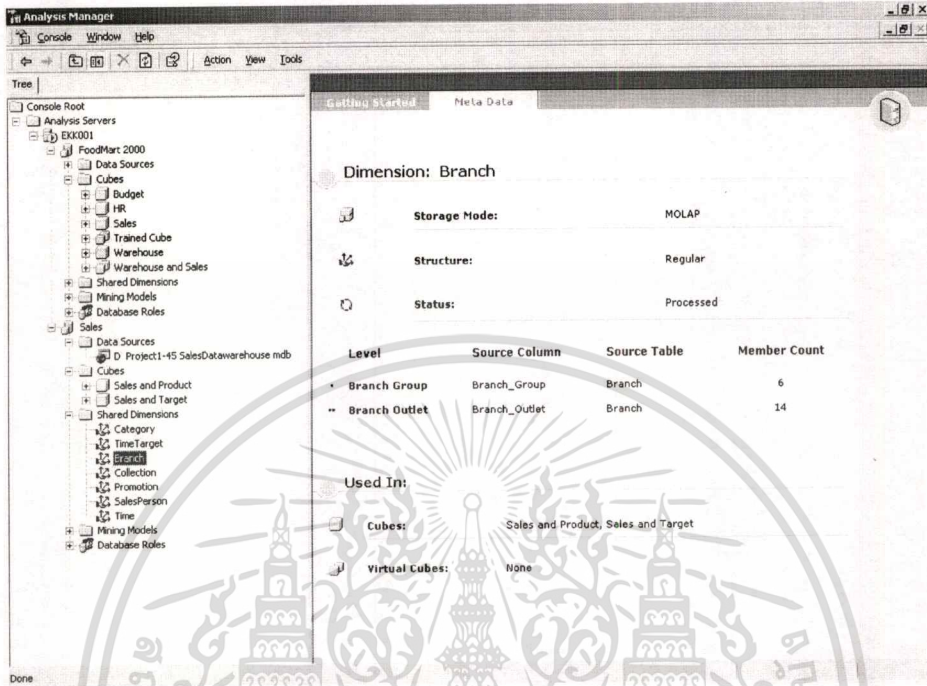
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● Metadata ของ Cube : Sales and Target

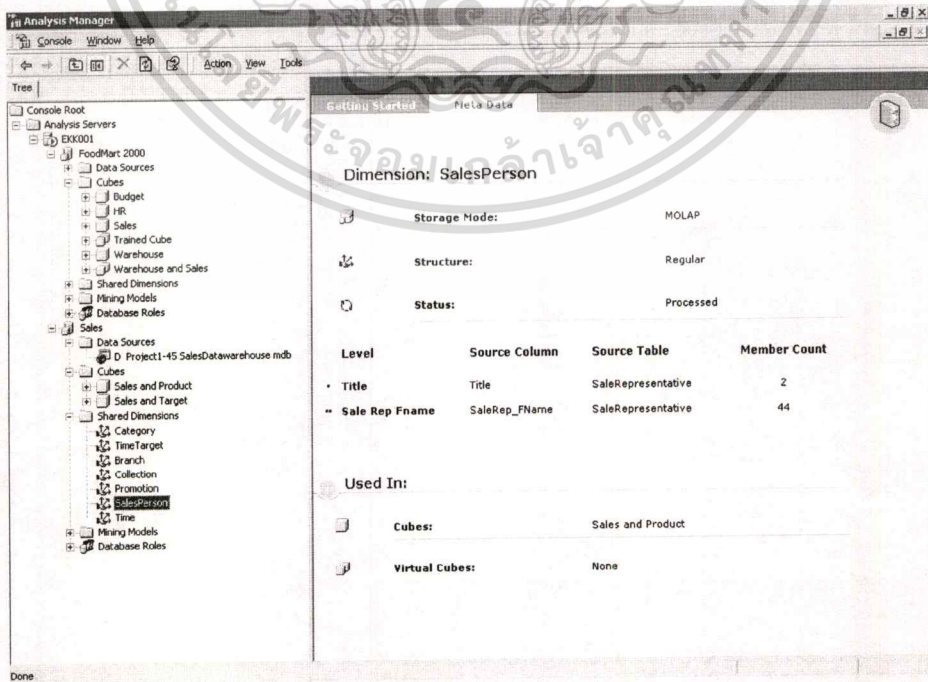


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● Metadata ของ Dimension : Branch

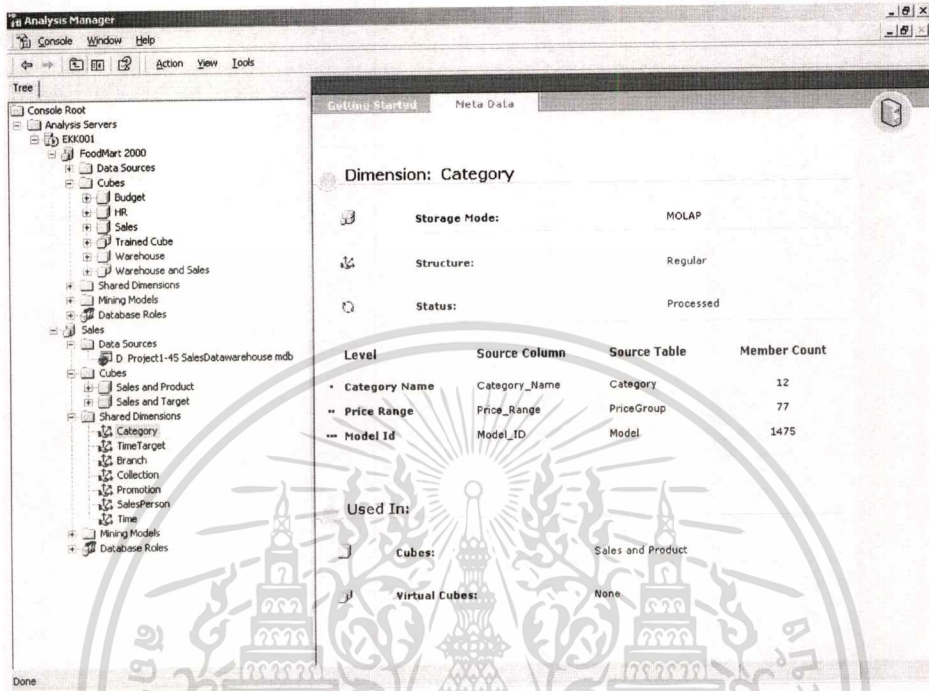


● Metadata ของ Dimension : SalesPerson

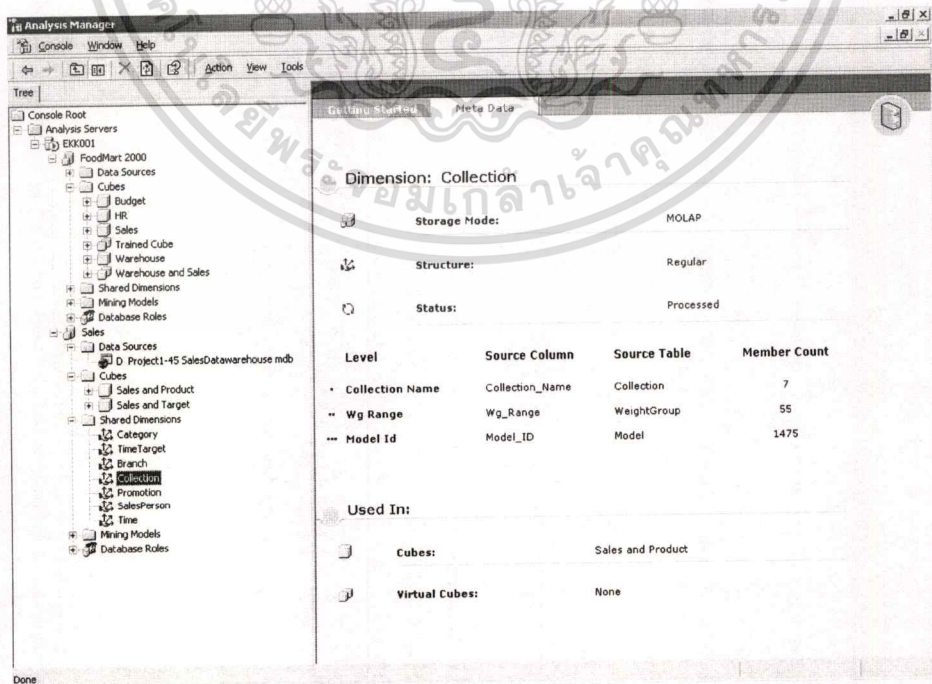


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● Metadata ของ Dimension : Category



● Metadata ของ Dimension : Collection



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ● Metadata ของ Dimension : Promotion

**Dimension: Promotion**

Storage Mode: MOLAP  
 Structure: Regular  
 Status: Processed

Level	Source Column	Source Table	Member Count
• Promotion Name	Promotion_Name	Promotion	7
•• Promotion Id	Promotion_ID	Promotion	10

Used In:

Cubes: Sales and Product  
 Virtual Cubes: None

## ● Metadata ของ Dimension : Time

**Dimension: Time**

Storage Mode: MOLAP  
 Structure: Regular  
 Status: Processed

Level	Source Column	Source Table	Member Count
• Year	DatePart('yyyy',SalesFact.Invoice_Date)	SalesFact	3
•• Quarter	DatePart('q',SalesFact.Invoice_Date)	SalesFact	11
••• Month	DatePart('m',SalesFact.Invoice_Date)	SalesFact	32

Used In:

Cubes: Sales and Product  
 Virtual Cubes: None

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Metadata ของ Dimension : TimeTarget

Dimension: TimeTarget

Storage Mode: MOLAP

Structure: Regular

Status: Processed

Level	Source Column	Source Table	Member Count
• Year	DatePart('yyyy', SaleTargetFact.Month)	SaleTargetFact	3
•• Quarter	DatePart('q', SaleTargetFact.Month)	SaleTargetFact	11
••• Month	DatePart('m', SaleTargetFact.Month)	SaleTargetFact	32

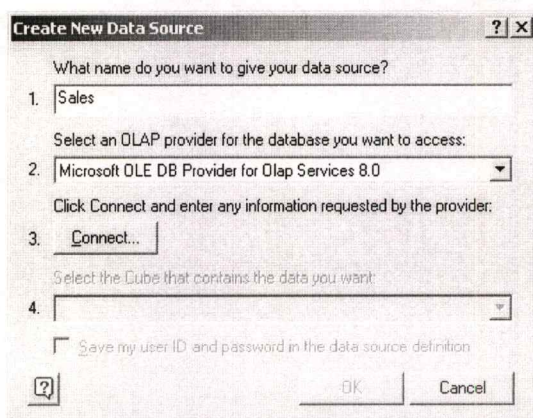
Used In:

Cubes: Sales and Target

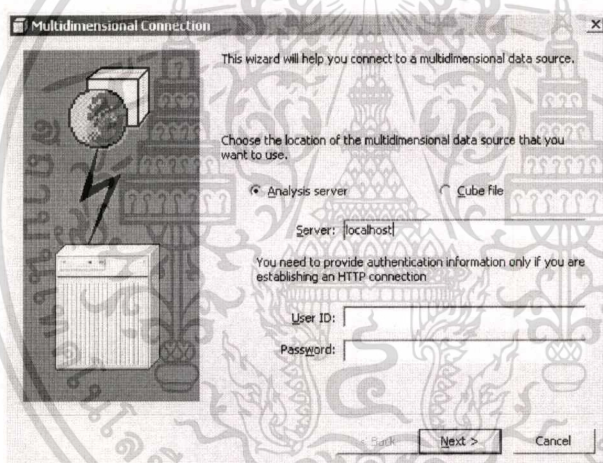
Virtual Cubes: None

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

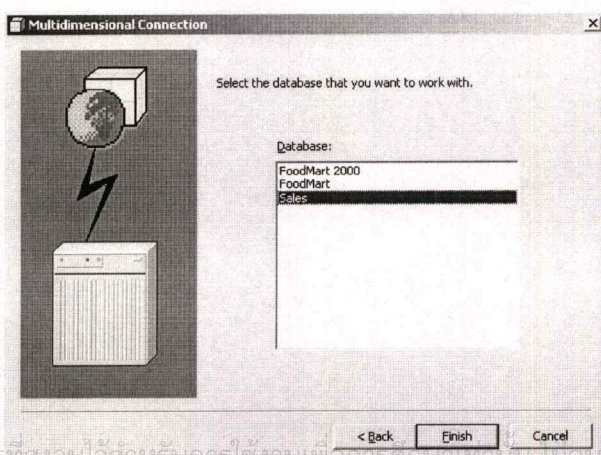




5. จะปรากฏกรอบโต้ตอบ Multidimensional Connection ที่พิมพ์ชื่อเครื่องคอมพิวเตอร์ที่เป็นเซิร์ฟเวอร์ หรือพิมพ์คำว่า localhost ถ้าเป็นการใช้งานอยู่บนเครื่องเซิร์ฟเวอร์

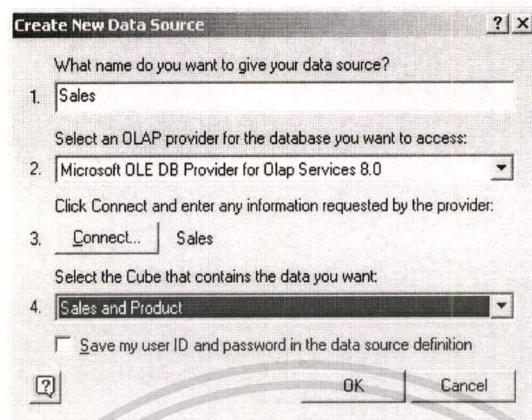


6. เลือกฐานข้อมูลที่ต้องการเชื่อมต่อ ซึ่งขณะนี้เราได้กำหนดการเชื่อมต่อกับฐานข้อมูลเซิร์ฟเวอร์แล้ว

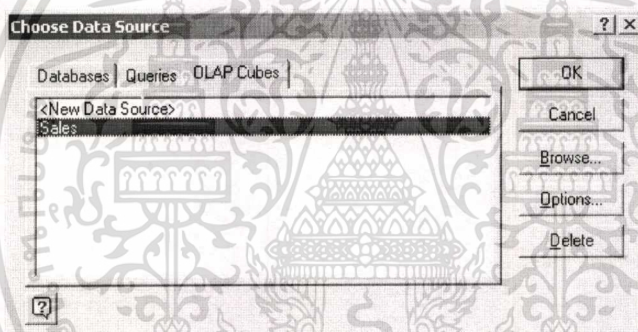


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. หากในฐานข้อมูลที่เชื่อมต่อมีคิวบ์มากกว่าหนึ่งคิวบ์ ให้ทำการเลือกคิวบ์ที่ต้องการ



8. แหล่งข้อมูลที่เราตั้งชื่อไว้จะแสดงขึ้นมาในรายการของคิวบ์แบบ OLAP



9. เลือกแหล่งข้อมูลที่ได้ทำการเชื่อมต่อ แล้วคลิก OK Microsoft Query จะดำเนินการกำหนดแหล่งข้อมูล OLAP Query ไปที่ Excel เข้าสู่ขั้นที่ 2-3 ตามปกติของการใช้ PivotTable and PivotChart Wizard

10. แถบเครื่องมือของ PivotTable จะมีปุ่มสำหรับโดเมนชั้นและ Measure ต่างๆ ซึ่งเป็นหัวใจในการสร้างรายงาน Excel PivotTable เมื่อมีปุ่มปรากฏขึ้นมาบนแถบเครื่องมือ นั่นคือเราได้เชื่อมต่อต่อกับ Analysis Server แล้ว

## ประวัติผู้เขียน

ชื่อ-สกุล	นางสาวบุษกร กนกบรรณกร
วันที่เกิด	13 กันยายน 2517
สถานที่เกิด	ระยอง
การศึกษา	วิทยาศาสตร์บัณฑิต (วัสดุศาสตร์) มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร
การทำงาน	พนักงานบริษัทโกลด์มาสเตอร์ จำกัด (มหาชน)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้